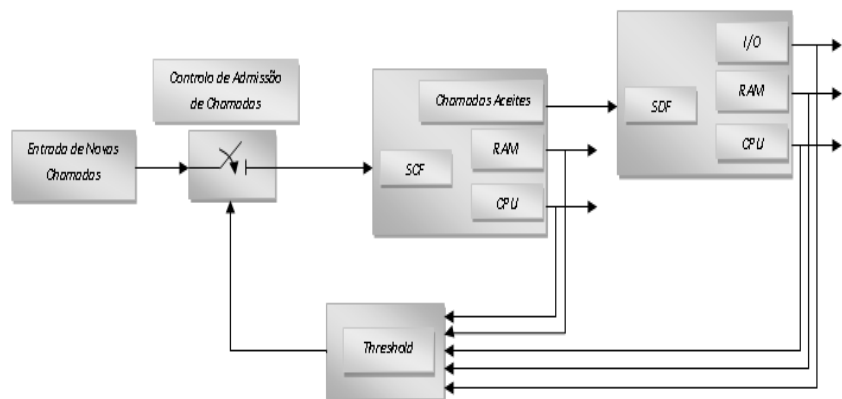


Marco António da
Costa Gonçalves

RECONFIGURAÇÃO AUTOMÁTICA DE REDES INTELIGENTES





**Marco António da
Costa Gonçalves**

RECONFIGURAÇÃO AUTOMÁTICA DE REDES INTELIGENTES

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e de Telecomunicações, realizada sob a orientação científica do Dr. Luís Almeida, Professor do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

Dissertação Elaborada em Parceria
com a Portugal Telecom Inovação.

Dedico este trabalho a todos os que me deram força e me ajudaram nos momentos difíceis e principalmente aos meus pais por todo o apoio, pela dedicação que me deram e pelo espírito de sacrifício que tiveram em todos estes anos da minha vida. A eles lhes devo inteiramente a pessoa que sou hoje.

O júri

Presidente

Doutor Aníbal Manuel de Oliveira Duarte
Professor Catedrático da Universidade de Aveiro

Doutor Paulo José Lopes Machado Portugal
Professor Auxiliar do Departamento de Engenharia Electrotécnica e de Computadores da
Faculdade de Engenharia da Universidade do Porto

Doutor Luís Miguel Pinho de Almeida
Professor Auxiliar da Universidade de Aveiro

Agradecimentos

Agradeço ao Professor Dr. Luís Almeida por todo o empenho e apoio nas mais diversas áreas desta dissertação, a todas as pessoas da PT Inovação envolvidas no projecto da Reconfiguração de Redes Inteligentes, em particular ao Eng. Miguel Santos por toda a orientação e ajuda na recolha de dados fundamentais para este trabalho e a José Ribeiro pela disponibilidade e ajuda em toda a parte de implementação prática.

Palavras-chave

Redes Inteligentes, Reconfiguração, Controlo de Carga, Controlador, CPU, RAM, Chamadas por Segundo (CPS), Limiar.

Resumo

O Conceito de “Redes Inteligentes (*Intelligent Network*)” corresponde a uma nova arquitectura de rede que surge com a crescente procura do operador de novos serviços e de maior flexibilidade do serviço telefónico, com possibilidade de personalizar o perfil de cada um desses serviços. Esta arquitectura assenta na separação do suporte lógico do serviço por um lado, e das funções de processamento e transporte das chamadas por outro. Deste modo consegue-se que a introdução destes serviços suplementares, o seu controlo e a sua gestão, seja conseguida de um modo rápido, eficaz e económico.

A reconfiguração dos sistemas de suporte às redes inteligentes é muito importante pois permite um melhor aproveitamento dos recursos do sistema e como tal garante um melhor desempenho e estabilidade de comportamento.

Assim, o projecto “Reconfiguração automática de redes inteligentes” visou o estudo e desenvolvimento de metodologias que procedam à reconfiguração automática desses sistemas de tempo real com base em requisitos considerados importantes para o bom desempenho e estabilidade dos sistemas de suporte a essas redes inteligentes.

Esta Dissertação focou no aspecto particular do controlo de chamadas para obter um controlo adequado de carga no sistema em termos de *CPU* e *RAM*. Numa primeira fase construiu-se um simulador que permitiu estudar o desempenho de várias técnicas de controlo realimentado. Entre essas técnicas deu-se especial relevo ao controlo *Proporcional* e *Proporcional-Integrador*, uma vez que são muito simples de implementar e apresentam baixo *overhead*. Finalmente, o controlo de chamadas desenvolvido foi efectivamente instalado no sistema real, dentro de uma zona estanque criada para testes.

Neste contexto, a técnica de controlo que se revelou mais eficaz para minimizar o número de chamadas rejeitadas em situações de sobrecarga é a que utiliza um limiar máximo de entrada de novas chamadas e um limiar mínimo dinâmico controlado por um controlo *Proporcional-Integrador*.

Keywords

Intelligent Networks, Reconfiguration, Load Control, Controller, CPU, RAM, Calls per Second (CPS), Threshold.

Abstract

The “Intelligent Networks” concept is a new architecture that emerged with the increasing search for new services and a bigger flexibility of the telephony service, with the possibility to personalize the profile of each one of these services. This architecture lays down on the separation of the logic support from the processing function service and call transport. This way, the introduction of additional services and its control is achieved in a faster, efficient and economic way.

The reconfiguration of the intelligent networks support system is very important because it allows improving the system resources granting a better performance and behavior stability.

Therefore, the “Intelligent Networks Reconfiguration”, project promotes the study and development of methodologies that provide support to the automatic reconfiguration of these real-time systems based on important requisites for the optimal performance and stability of these intelligent networks support systems.

This dissertation focused on the particular aspect of call admission control to obtain a proper load control of *CPU* and *RAM* on the system. In a first step, a simulator was built that allowed to study the performance of different *feedback* control techniques. Among these techniques special attention was given to the *Proportional* and *Proportional-Integrator* control because they are very simple to implement and present a low *overhead*. Finally, the developed load control was effectively installed in the real system, in an isolated module created for tests.

In this context, the technique that revealed itself more efficient to reduce the number of rejected calls in overload situations is the technique that uses a static maximum *Threshold* of new arriving calls, and a dynamic minimum *Threshold* controlled by a *Proportional-Integrator* controller.

Conteúdo

Lista de Acrónimos	9
Índice de Tabelas	11
Índice de Figuras.....	13
2. Enquadramento	17
2.1 Origem das Redes Inteligentes	17
2.2 Características Básicas da Arquitectura IN	20
2.3 Arquitectura IN.....	20
2.4 Estratégia para a Introdução da Rede Inteligente	23
2.5 Necessidade de Reconfiguração Automática da RI	24
3. Estudo de Algoritmos de Controlo de Carga	25
3.1 Modelização do Sistema	25
3.3 Simulação sem Controlo.....	29
3.3.1 Simulação de um <i>SCF</i>	30
3.3.2 Simulação de um <i>SCF</i> e um <i>SDF</i>	30
3.3.3 Simulação de quatro <i>SCF</i> e um <i>SDF</i>	31
3.4 Simulação do Controlo com um <i>Limiar</i>	32
3.4.1 Controlo de um <i>SCF</i>	33
i) Um Limiar Estático	33
ii) Um Limiar Dinâmico com Controlo Proporcional.....	34
iii) Um Limiar Dinâmico com Controlo Proporcional-Integrador.....	37
3.4.2 Controlo de um <i>SCF</i> e um <i>SDF</i>	38
3.4.3 Controlo de quatro <i>SCF</i> e um <i>SDF</i>	42
3.5 Simulação do Controlo com dois <i>Limiares</i>	46
3.5.1 Controlo de um <i>SCF</i>	46
3.5.2 Controlo de um <i>SCF</i> e um <i>SDF</i>	47
3.5.3 Controlo de quatro <i>SCF</i> e um <i>SDF</i>	49
3.6 Controlo com um <i>Limiar Estático</i> e um <i>Limiar Dinâmico</i>	52
3.6.1 Controlo de um <i>SCF</i>	52
3.6.2 Controlo de um <i>SCF</i> e um <i>SDF</i>	53
3.6.3 Controlo de quatro <i>SCF</i> e um <i>SDF</i>	55
3.7.1 Controlo de um <i>SCF</i>	58
3.7.2 Controlo de um <i>SCF</i> e um <i>SDF</i>	59
3.7.3 Controlo de quatro <i>SCF</i> e um <i>SDF</i>	61
3.8 Simulação do Controlo com Janela Temporal	64
3.8.1 Controlo de um <i>SCF</i>	64
3.8.2 Controlo de um <i>SCF</i> e um <i>SDF</i>	65
3.8.3 Controlo de quatro <i>SCF</i> e um <i>SDF</i>	67

3.9 Discussão de Resultados.....	70
4. Implementação.....	73
4.1 Sistema Real	73
4.1 Controlo de Carga no <i>SCF</i> (Cliente)	75
4.2 Controlo de Carga no <i>SDF</i> (Servidor)	79
5. Conclusão e Trabalho Futuro	83
5.1 Conclusão	83
5.2 Trabalho Futuro	83
Referências	85

Lista de Acrónimos

SCP	Service Control Points
SCF	Service Control Function
SDP	Service Data Points
SDF	Service Data Function
CPS	Calls per Second
SMS	Service Management System
SSP	Service Switching Point
SSF	Service Switching Function
CCF	Control Call Function
IP	Intelligent Peripheral
SCE	Service Creation Environment
SCEF	Service Creation Environment Function
SRF	Specialized Resource Function
PI	Proporcional-Integrador
IN	Intelligent Networks
IR	Redes Inteligentes

Índice de Tabelas

<i>Tabela 1 - Cronologia das Redes Inteligentes [1, 9, 10, 11].</i>	19
<i>Tabela 2 - Valores Reais e Estimados de SCF.</i>	26
<i>Tabela 3 - Resultado da Simulação do controlo do SCF com um Limiar Estático.</i>	34
<i>Tabela 4 - Resultados da Simulação do Controlo do SCF com um Limiar Dinâmico Proporcional.</i>	35
<i>Tabela 5- Resultados da Simulação do Controlo do SCF com um Limiar Dinâmico PI.</i>	38
<i>Tabela 6- Resultados do Controlo do SCF e SDF (um Limiar).</i>	40
<i>Tabela 7 - Pesos dos SCF em cada nível.</i>	43
<i>Tabela 8 – SCF: Resultados da Simulação com quatro SCF e um SDF.</i>	43
<i>Tabela 9 - SDF: Resultados da Simulação com quatro SCF e um SDF.</i>	45
<i>Tabela 10 - Resultados do Controlo de um SCF.</i>	46
<i>Tabela 11 - Resultados do Controlo do SCF e SDF (dois Limiares).</i>	48
<i>Tabela 12 - SDF: Resultados da Simulação com quatro SCF e um SDF (dois Limiares).</i>	49
<i>Tabela 13 - SCF: Resultados da Simulação com quatro SCF e um SDF (dois Limiares).</i>	50
<i>Tabela 14 - Resultados do Controlo de um SCF (um Limiar Estático e um Limiar Dinâmico).</i>	52
<i>Tabela 15 - Resultados do Controlo do SCF e SDF (um Limiar Estático e um Limiar Dinâmico).</i>	53
<i>Tabela 16 - SCF: Resultados da Simulação com quatro SCF e um SDF (um Limiar Estático e um Limiar Dinâmico).</i>	56
<i>Tabela 17- SDF: Resultados da Simulação com quatro SCF e um SDF (um Limiar Estático e um Limiar Dinâmico)</i>	57
<i>Tabela 18 - Resultados do Controlo de um SCF (um Limiar Eventos).</i>	58
<i>Tabela 19 - Resultados do Controlo do SCF e SDF (Limiar Eventos).</i>	60
<i>Tabela 20 - SDF: Resultados da Simulação com quatro SCF e um SDF (Limiar Eventos).</i>	61
<i>Tabela 21 - SCF: Resultados da Simulação com quatro SCF e um SDF (Limiar Eventos).</i>	62
<i>Tabela 22 - Resultados do Controlo de um SCF (Janelas Temporais).</i>	64
<i>Tabela 23 - Resultados do Controlo do SCF e SDF (Janelas Temporais).</i>	66
<i>Tabela 24 - SDF: Resultados da Simulação com quatro SCF e um SDF (Janelas Temporais).</i>	68
<i>Tabela 25 - SCF: Resultados da Simulação com quatro SCF e um SDF (Janelas Temporais).</i>	69
<i>Tabela 26 - Ordenação dos Algoritmos de Controlo em Relação ao menor N° de Rejeição (um SCF).</i>	71
<i>Tabela 27 - Ordenação dos Algoritmos de Controlo em Relação ao menor N° de Rejeição (um SCF e um SDF).</i>	71
<i>Tabela 28- Ordenação dos Algoritmos de Controlo em Relação ao menor N° de Rejeição (quatro SCF e um SDF).</i>	71

Índice de Figuras

<i>Figura 1 - Evolução da Normalização, ITU-T e ETSI [9].</i>	18
<i>Figura 2 – Principais Unidades Funcionais [1].</i>	21
<i>Figura 3 – Relações funcionais e pontos de referência para o CS-1 [1].</i>	22
<i>Figura 4 - Relações funcionais e pontos de referência para o CS-2 [9].</i>	23
<i>Figura 5 - Modelização do Sistema.</i>	25
<i>Figura 6 - Esquema do Controlo com quatro SCF.</i>	27
<i>Figura 7 - Geradores de Chamadas e Respetivos Valores.</i>	28
<i>Figura 8 - Geração de Chamadas.</i>	28
<i>Figura 9 - N° de chamadas que chegam por hora ao longo de 1 Dia.</i>	29
<i>Figura 10 - Simulação sem controlo de um SCF.</i>	30
<i>Figura 11 - Simulação sem controlo do SCF.</i>	30
<i>Figura 12 - Simulação sem controlo do SDF.</i>	31
<i>Figura 13 - Simulação sem controlo dos quatro SCF.</i>	31
<i>Figura 14 - Simulação sem controlo do SDF com influência dos quatro SCF.</i>	32
<i>Figura 15 - Esquema de Aceitação e Rejeição do SCF.</i>	32
<i>Figura 16 - Controlo com um Limiar.</i>	33
<i>Figura 17 - Simulação do controlo do SCF com um Limiar Estático.</i>	33
<i>Figura 18 - Malha de Controlo do SCF com Limiar Dinâmico</i>	34
<i>Figura 19 - Simulação do Controlo do SCF com um Limiar Dinâmico Proporcional.</i>	36
<i>Figura 20 - Simulação do Controlo do SCF com um Limiar Dinâmico PI.</i>	38
<i>Figura 21 – Malha de Controlo de um SCF e de um SDF com Limiar Dinâmico.</i>	39
<i>Figura 22 – SCF: Simulação com Controlo PI no SCF e SDF (melhor resultado).</i>	41
<i>Figura 23 - SDF: Simulação com controlo PI no SCF e SDF (melhor resultado).</i>	41
<i>Figura 24 - Malha de Controlo de quatro SCF e de um SDF com Limiar Dinâmico.</i>	42
<i>Figura 25 - SCF: Simulação com Controlo PI nos quatro SCF e no SDF (melhor resultado).</i>	44
<i>Figura 26 - SDF: Simulação com Controlo PI nos quatro SCF e no SDF (melhor resultado).</i>	44
<i>Figura 27 - Esquema de Aceitação e Rejeição do SCF (dois Limiares).</i>	46
<i>Figura 28 - Simulação do Controlo do SCF com dois Limiares Dinâmicos PI (melhor resultado).</i>	47
<i>Figura 29 - SCF: Simulação do Controlo do SCF e SDF com dois Limiares Dinâmicos PI (melhor resultado).</i>	47
<i>Figura 30 - SDF: Simulação do Controlo do SCF e SDF com dois Limiares Dinâmicos PI (melhor resultado).</i>	48
<i>Figura 31 - SDF: Simulação com Controlo PI nos quatro SCF e no SDF (dois Limiares: melhor resultado).</i>	51
<i>Figura 32 - SDF: Simulação com Controlo PI nos quatro SCF e no SDF (dois Limiares: melhor resultado)</i>	51
<i>Figura 33 - Simulação do Controlo do SCF com um Limiar Estático e um Limiar Dinâmico PI (melhor resultado).</i>	52
<i>Figura 34 - SCF: Simulação do Controlo do SCF e SDF com um Limiar Estático e um Limiar Dinâmico PI (melhor resultado).</i>	54
<i>Figura 35 - SDF: Simulação do Controlo do SCF e SDF com um Limiar Estático e 1 Limiar Dinâmico PI (melhor resultado).</i>	54
<i>Figura 36 - SCF: Simulação com Controlo PI nos quatro SCF e no SDF (um Limiar Estático e um Limiar Dinâmico).</i>	55
<i>Figura 37 - SDF: Simulação com Controlo PI nos quatro SCF e no SDF (um Limiar Estático e um Limiar Dinâmico).</i>	55
<i>Figura 38- Esquema de Aceitação e Rejeição do SCF (Limiar eventos).</i>	58
<i>Figura 39 - Simulação do Controlo do SCF com um Limiar Eventos PI (melhor resultado).</i>	59
<i>Figura 40 - SCF: Simulação do Controlo do SCF e SDF com um Limiar Eventos Dinâmico PI (melhor resultado).</i>	59
<i>Figura 41 - SDF: Simulação do Controlo do SCF e SDF com um Limiar Eventos Dinâmico PI (melhor resultado).</i>	60
<i>Figura 42 – SCF: Controlo PI nos quatro SCF e no SDF com um Limiar Eventos Dinâmico (melhor resultado).</i>	63
<i>Figura 43 - SDF: Controlo PI nos quatro SCF e no SDF com um Limiar Eventos Dinâmico (melhor resultado).</i>	63

<i>Figura 44 – Esquema de Controlo com Janelas Temporais.....</i>	<i>64</i>
<i>Figura 45 - Simulação do Controlo do SCF PI Janela Temporal (melhor resultado).</i>	<i>65</i>
<i>Figura 46 - SCF: Simulação do Controlo do SCF e SDF PI Janelas Temporais (melhor resultado).</i>	<i>66</i>
<i>Figura 47 - SDF: Simulação do Controlo do SCF e SDF PI Janelas Temporais (melhor resultado).</i>	<i>67</i>
<i>Figura 48 - SDF: Simulação com Controlo PI nos quatro SCF e no SDF Janelas Temporais (melhor resultado).</i>	<i>68</i>
<i>Figura 49 - SCF: Simulação com Controlo PI nos quatro SCF e no SDF Janelas Temporais (melhor resultado).</i>	<i>69</i>
<i>Figura 50 – Arquitectura do Sistema Real.</i>	<i>73</i>
<i>Figura 51 - Malha de Controlo do Sistema Real.....</i>	<i>74</i>
<i>Figura 52 - Diagrama de Funcionamento de função getPermissiom.</i>	<i>76</i>
<i>Figura 53- Arranque do SCF.</i>	<i>77</i>
<i>Figura 54 - Log doSCF.</i>	<i>77</i>
<i>Figura 55 - Resultados dos testes do Algoritmo de Controlo no SCF.....</i>	<i>78</i>
<i>Figura 57 - Diagrama de Funções da Classe LoadControlServerCls.</i>	<i>79</i>
<i>Figura 56 - Diagrama de Funções da Classe System.</i>	<i>79</i>
<i>Figura 58 - Diagrama de Funções da Classe LoadControlGroupCls.</i>	<i>80</i>
<i>Figura 59 - Diagrama de Funções da Classe LoadControlServiceCls.</i>	<i>80</i>
<i>Figura 60 - Teste Simulado do Controlo no Servidor no SCF.</i>	<i>81</i>
<i>Figura 61 - Teste Simulado do Controlo no Servidor no SCF.</i>	<i>82</i>

1. Introdução

As redes de telecomunicações tradicionais utilizam diferentes tecnologias de diferentes fabricantes para fornecerem os seus serviços de comutação telefónica. Hoje em dia os operadores procuram cada vez mais novos serviços que exigem uma maior flexibilidade do serviço telefónico [1].

Numa rede tradicional a introdução destes serviços adicionais a uma escala global obriga à implementação desses serviços em cada um dos sistemas de comutação com a correspondente actualização em todos os comutadores. Assim este processo torna-se bastante demorado e envolve elevadíssimos custos de manutenção e configuração, além de que cada implementação destes serviços adicionais pode depender ainda dos fabricantes dos comutadores [1].

É neste cenário que surge o conceito de “Redes Inteligentes (*Intelligent Network*)”. As Redes Inteligentes (*RI*) correspondem a uma nova arquitectura de rede que assentam na separação do suporte lógico do serviço por um lado, e das funções de processamento e transporte das chamadas por outro. Deste modo consegue-se que a introdução destes serviços suplementares, o seu controlo e a sua gestão seja conseguida de um modo rápido, eficaz e económico [1,2].

As Redes Inteligentes são compostas, numa visão simplificada, pela rede de comutação propriamente dita e por um conjunto de elementos de controlo, entre os quais um ou vários *SCP* (*Service Control Points*) e *SDP* (*Service Data Points*). Os *SCP* fornecem a Lógica de Serviço das Redes Inteligentes executando uma ou várias *Service Control Functions* (*SCF*) que incluem o controlo de admissão de cada pedido de serviço, e.g., de cada chamada telefónica. Os *SDP* contêm os dados referentes a essa lógica de Serviço, como por exemplo, o modelo de taxação, o saldo, etc. Estes dados são disponibilizados através das *Service Data Functions* (*SDF*). Uma *SDF* pode conter os dados de controlo de várias *SCF* e podem residir na mesma ou em máquinas distintas.

Os *SCP* e os *SDP* são nós vitais para as Redes Inteligentes e podem ser acedidos por vários utilizadores situados em pontos diferentes. Numa arquitectura distribuída, como é o caso desta rede e onde cada utilizador actua de forma independente, poderemos encontrar facilmente situações de sobrecarga que podem criar o colapso de toda a Rede. Será preciso então desenvolver mecanismos de controlo de carga de forma assegurar a manutenção do bom funcionamento da rede [3].

A Reconfiguração dos sistemas de suporte às redes inteligentes vai permitir um melhor aproveitamento dos recursos físicos do sistema tal como uma melhor estabilidade e desempenho no comportamento deste. Pretende-se então estudar, melhorar e desenvolver metodologias que procedam à reconfiguração automática em tempo real do Sistema de Redes actualmente implementado, permitindo também um melhor aproveitamento dos recursos humanos. Esta reconfiguração vai servir para controlar a carga de chamadas admitidas pelo sistema garantindo níveis mínimos de qualidade de serviço desejado pelo operador, mesmo na presença de sobrecargas.

Esta Dissertação focou no aspecto particular do controlo de chamadas para obter um controlo adequado de carga no sistema em termos de *CPU* e *RAM* [3, 4, 5, 6]. Numa primeira fase construiu-se um simulador que permitiu estudar o desempenho de várias técnicas de controlo realimentado. Entre essas técnicas deu-se especial relevo ao controlo *Proporcional* e *Proporcional-Integrador* [7, 8], uma vez que são muito simples de implementar e apresentam baixo *overhead*. Finalmente, o controlo de chamadas desenvolvido foi efectivamente instalado no sistema real, dentro de uma zona estanque criada para testes.

Neste contexto, a técnica de controlo que se revelou mais eficaz para minimizar o número de chamadas rejeitadas em situações de sobrecarga é a que utiliza um limiar máximo de entrada de novas chamadas e um limiar mínimo dinâmico controlado por um controlo *Proporcional-Integrador*.

Esta dissertação encontra-se organizada com a seguinte estrutura:

Capítulo 2 – Exposição do enquadramento geral do trabalho, apresentando de uma forma mais detalhada as denominadas *Intelligent Networks*, o estado de arte e a sua respectiva evolução ao longo dos tempos.

Capítulo 3 – Simulação de um sistema de *Redes Inteligentes*. Estudo e análise de algoritmos de *Controlo de Carga* para a reconfiguração automática dessa *RI*.

Capítulo 4 – Detalhes da implementação prática efectuada.

Capítulo 5 – Conclusões finais e abordagem de melhorias ao sistema que poderão ser efectuadas em implementações futuras.

2. Enquadramento

2.1 Origem das Redes Inteligentes

Como já foi referido, os clientes actuais procuram por uma gama de serviços adicionais que consequentemente obrigam a que a rede telefónica apresente grande flexibilidade. A inserção de um novo serviço adicional numa rede telefónica tradicional levanta uma série de problemas e transtornos, tais como elevados custos de manutenção e configuração, processo de implementação demorado e dependência dos fabricantes dos comutadores. É neste contexto que surge o conceito de *Redes Inteligentes*. A arquitectura destas *Redes Inteligentes* assenta na separação do suporte lógico do serviço por um lado e das funções de processamento e transporte das chamadas por outro. Desta forma, a introdução de um novo serviço adicional e o seu respectivo controlo e gestão é conseguida de uma forma rápida, eficaz e consequentemente mais económica.

O conceito da separação do suporte lógico do serviço das funções de processamento e transporte das chamadas teve origem em 1984 nos EUA após a criação do “*Feature Node*” pelo operador regional *Ameritech* (uma das *RBOC – Regional Bell Operating Company*). Este conceito foi depois sendo desenvolvido pela *Bellcore* até que em 1986 passa a ser denominado de *Rede Inteligente (IN)* [1].

A primeira arquitectura de *Rede Inteligente*, conhecida como *IN/1*, teve origem na *Bellcore* e foi apresentada como um meio de adicionar inteligência à rede telefónica pública de um modo centralizado. A arquitectura da *IN/1* depende do tipo de serviço e é baseada numa aproximação de blocos de construção elementares. Nesta arquitectura a introdução de novos serviços implica alterações nas estações [1].

Em 1986 surge, também através da *Bellcore*, uma nova arquitectura conhecida *IN/2*. Esta nova arquitectura tinha por objectivo conseguir uma completa independência da lógica de serviço relativamente à arquitectura dos sistemas de comutação. A construção de serviços nesta arquitectura baseia-se num conjunto de elementos e serviços básicos denominados de componentes funcionais, tais como reencaminhamento de chamadas, identificação automática da origem da chamada, podendo estes componentes funcionais ser combinados para construção de diferentes serviços de valor acrescentado [1].

A arquitectura da *IN/2* apresentava alguns problemas de implementação em tempo útil dada a sua complexidade. Para colmatar este problema foi definida uma arquitectura intermédia denominada de *IN/1+* que deveria estar implementada entre 1990-1991. No entanto, não estavam criadas as condições necessárias, especialmente por parte dos fabricantes, para a implementação destas duas arquitecturas, por isso foram ambas abandonadas. Apercebendo-se da necessidade de cooperação entre os diferentes intervenientes neste domínio incluindo os fabricantes de sistemas de telecomunicação, *Bellcore* organizou em 1989 um fórum aberto, o “*Multi-Vendor Interactions*” (*MVI*), que

veio definir uma nova arquitectura funcional e física denominada de “*Advanced Intelligent Network*” (AIN).

Em 1994 alguns países possuíam em operação arquitecturas do tipo IN/1, em que foram explorados os serviços de Número Verde ou de Validação de Cartões (de crédito ou outros). Nestes países, houve também uma transição progressiva para arquitecturas independentes do serviço [1].

O trabalho de normalização destas arquitecturas tem sido feito desde 1989 pelo *Instituto Europeu de Normalização em Redes Inteligentes (ETSI)*, através da sub-comissão técnica NA6. Todo este trabalho de normalização foi sendo sempre desenvolvido em estreita ligação com o *Organismo de Normalização Internacional (ITU-T: grupo de estudo XI.4)* [1].

Desde então estas redes tem sido cada vez mais normalizadas, em geral, como suporte às novas solicitações criadas pelas comunicações móveis, tendo sido criado mais alguns Organismos de Normalização, como por exemplo o *3GPP*, que foi criado em Dezembro de 1998 e apresenta parceiros organizacionais como *ARIB*, *CCSA*, *ETSI*, *ATIS*, *TTA* e *TTC* [10]. Em termos técnicos, contudo, não se verificaram grandes mudanças na arquitectura física da rede, tendo as modificações ocorridas sido sobretudo no plano do suporte lógico do serviço das funções de processamento de chamadas.

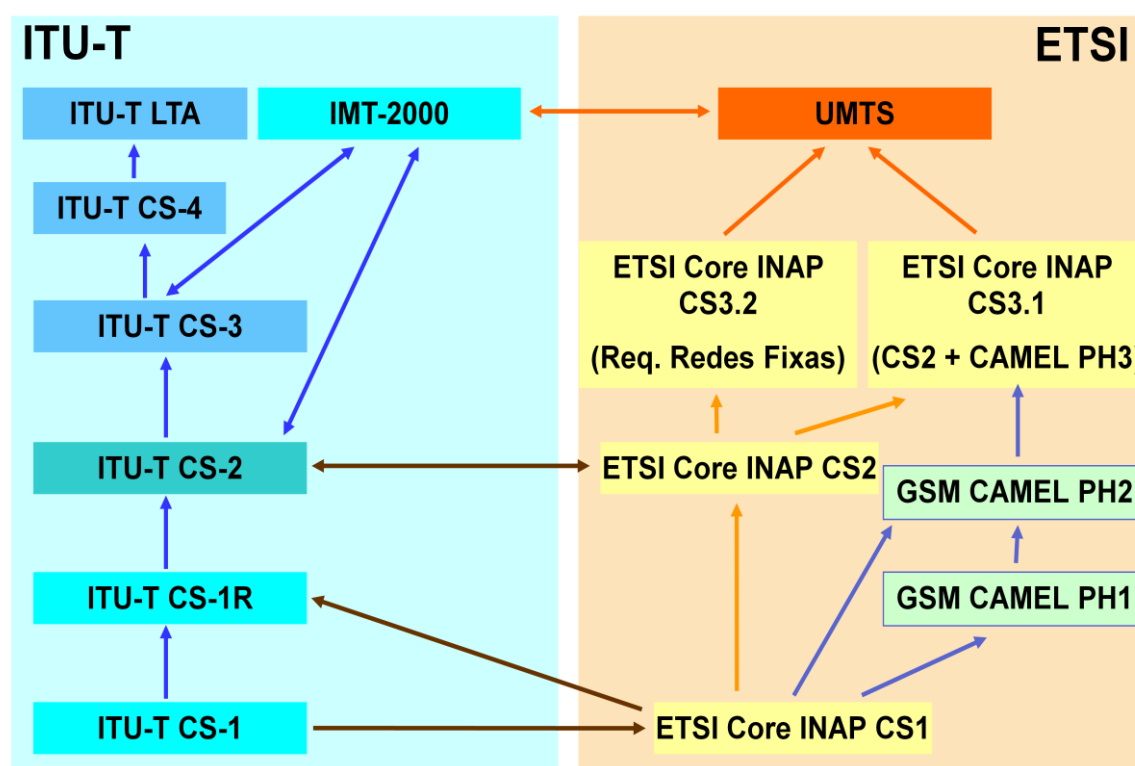


Figura 1 - Evolução da Normalização, ITU-T e ETSI [9].

1984	Conceito IN/1 na Bellcore	<ul style="list-style-type: none"> • Bases de dados centralizadas concentradoras da inteligência da rede e consultadas pelos comutadores • Aproximação aos blocos construtores de serviços • Arquitectura dos comutadores dependentes dos serviços
1986	Conceito IN/2 na Bellcore	<ul style="list-style-type: none"> • Lógica do serviço independente dos comutadores • Construção dos serviços baseada nas componentes funcionais (ex: reencaminhamento da chamada, identificação automática da origem da chamada) • Complexidade da arquitectura
1988	IN/1+ na Bellcore	<ul style="list-style-type: none"> • IN/2 simplificada • Objectivo: implementação até 1991
1989	MVI – Multivendors Interaction	<ul style="list-style-type: none"> • Fórum entre Bellcore e fabricantes • AIN – Advanced IN • ITU-T • Início da especificação das normas IN no grupo XI • Capabilities Sets (CS) evolutivos CS1 → CS2 → CSN • ETSI • Início da especificação no NA& seguindo-se NA!, NA7, SPS1, SPS2, SPS3
1992	ITU-T	<ul style="list-style-type: none"> • Publica a primeira versão das normas relativas ao CS1
1994	ETSI	<ul style="list-style-type: none"> • Publica as normas CORE INAP baseadas no CS1
1996	ETSI	<ul style="list-style-type: none"> • GSM CAMEL Fase 1 • GSM CAMEL Fase 1 + R96
1997	ETSI	<ul style="list-style-type: none"> • GSM CAMEL Fase 2 • GSM CAMEL Fase 2 + R97
1998	ETSI / ITU-T / 3GPP	<ul style="list-style-type: none"> • GSM CAMEL Fase 2 + R98 (ETSI) • Criação do 3GPP • Início da definição da rede 3G sob a designação de IMT-2000 (International Mobile Telephony 2000), cuja outra designação equivalente é UMTS (Universal Mobile Telecommunications Systems) (ITU-T)
1999	3GPP	<ul style="list-style-type: none"> • CAMEL Fase 3 • 3G R99 (IMT-200, UMTS)
2000	3GPP	<ul style="list-style-type: none"> • 3G Rel-4
2002	3GPP	<ul style="list-style-type: none"> • CAMEL Fase 4 • 3G Rel-5
2003	3GPP	<ul style="list-style-type: none"> • 3G Rel-6
2004	3GPP	<ul style="list-style-type: none"> • Melhoramento na Rede 3G Rel-6
2005	3GPP	<ul style="list-style-type: none"> • 3G Rel-7
2006	3GPP	<ul style="list-style-type: none"> • Melhoramentos na Rede 3G Rel-7

Tabela 1 - Cronologia das Redes Inteligentes [1, 9, 10, 11].

2.2 Características Básicas da Arquitectura IN

Uma arquitectura *IN* apresenta as seguintes características básicas [1, 9]:

- Inteligência sobre o controlo das ligações na rede concentrada em nós centralizados. Estes nós da rede são designados por *Pontos de Controlo do Serviço (SCP)*.
- Nós de redes responsáveis pelo estabelecimento das ligações na rede sob a direcção do *SCP*. Estes nós da rede são designados por *Pontos de Comutação do Serviço (SSP)*.
- Capacidades de criação de serviços, que permitam a criação rápida e económica de novos serviços. Esta capacidade permitirá ao Operador de Telecomunicações identificar a necessidade de um dado serviço, criar este serviço, simular o seu funcionamento na rede e introduzi-lo na rede na altura adequada, mantendo a integridade da mesma rede.
- Capacidade de permitir a adaptação dos diferentes serviços aos requisitos individuais dos clientes dos mesmos.
- Interfaces normalizados entre os componentes da arquitectura, nomeadamente entre o *SCP* e o *SSP*, para a normalização da comunicação entre a lógica do serviço e a rede de telecomunicações, e entre o *SCP* e o ambiente de criação de serviços (*SMS/SCE*). Estes interfaces normalizados irão facilitar a competição entre os fornecedores de equipamentos para a rede, tornando assim os Operadores de Rede mais independentes das soluções proprietárias dos fabricantes.

2.3 Arquitectura IN

A arquitectura das *Redes Inteligentes* assenta, como já foi referido várias vezes, na separação do suporte lógico do serviço das funções de processamento e transporte das chamadas. Este conceito permite então ao operador de rede uma maior flexibilização no mapeamento de entidades físicas e funcionais [1].

SSP – Ponto de Comutação do Serviço (Service Switching Point)

O *SSP* é um nó responsável pelo estabelecimento das ligações físicas através da rede de transporte e controlado pelo *SCP*. Contém a função *SSF (Service Switching Function)* que faz o reconhecimento das chamadas utilizando serviços da *Rede Inteligente*. A *SSF* vai interagir com a *CCF (Control Call Function)* de modo a que a *SCF* tenha uma visão lógica da chamada permitindo o controlo do estabelecimento das ligações. As chamadas *IN* que não são reconhecidas pelos sistemas de comutação, são encaminhadas para os *SSP* que os servem.

SMS - Service Management System

SCE - Service Creation Environment

SCP - Service Control Point

SSP - Service Switching Point

IP - Intelligent Peripheral

PSTN - Public Switch Telephone Network

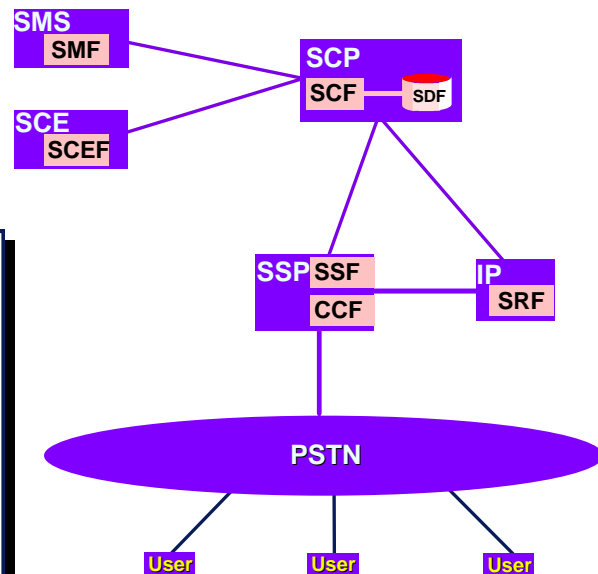


Figura 2 – Principais Unidades Funcionais [1].

SMS – Sistema de Gestão do Serviço (Service Management system)

O *SMS* fornece funções de gestão dos serviços. Controla as actividades de gestão, administração e manutenção associadas aos serviços em cada um dos elementos da rede (*SCP*, *IP*, *SSP*). O *SMS* faz também o controlo de admissão dos clientes dos serviços para fins de configuração e interrogação de modo a poder providenciar um ambiente de engenharia de software para a criação de novos serviços.

O *SMS* é um *Sistema de Operação* (*OS – Operating System*) especializado em funções de gestão de serviços de uma *Rede de Gestão de Telecomunicações* (*TMN – Telecommunications Management Network*). A entidade funcional *SMS* (*Service Management Functions*) é incluída dentro do *SMS* e tem como função fazer o interface com virtualmente qualquer outra entidade funcional da rede para a transferência de informação relacionada com a lógica e de dados do serviço.

SCE – Ambiente de Criação de Serviços (Service Creation Environment)

A verificação e teste de novos serviços de *Rede Inteligente* e a capacidade de definição são proporcionadas pela função *SCEF* (*Service Creation Environment Function*) que é suportada pelo *SCE*. Esta função vai fornecer a lógica do serviço para ser depois instalada pela função *SMF* nos vários *SCP*.

IP – Periférico Inteligente (Intelligent Peripheral)

O *IP* contém diversas funções úteis em diversos serviços *IN*, que de outra forma teriam que residir em todos os *SSP*. Essas funções úteis podem ser funções como reconhecimento de mensagens de voz e os anúncios de mensagens. A função *SRF* (*Specialized ResourcePeripheral*) encontra-se dentro do *Periférico Inteligente* e é responsável por toda a interacção da *IN* com o seu utilizador ou cliente.

SCP – Ponto de Controlo do Serviço (Service Control Point)

O *SCP* é um nó de rede centralizado e tem como função concentrar a “inteligência” da rede e guardar informações relativas ao serviço e o suporte lógico do serviço, fornecendo ao *SSP* os dados necessários para completar uma tentativa de chamada. O *SCP* pode interligar-se por duas vias, directamente ao *SSP* por intermédio de uma interface particular, sendo esta configuração denominada de *SCP-Adjunto*, ou pode ligar-se por meio de uma interface normalizada servindo desta maneira mais de que um *SSP*. A lógica de serviço do *SCP* encontra-se na função *SCF* (*Service Control Function*) e fornece todo o controlo lógico aplicado a uma chamada. Pode-se encontrar também no *SCP* em alguns casos, função *SDF* (*Service Data Function*) que fornece à função *SCF* uma visão lógica dos dados relacionados com o serviço e com a rede.

Existem outras duas entidades bastante importantes na arquitectura da *Rede Inteligente*, são a *Rede de Sinalização Nº 7* do ITU-T (*SSN7*), que é utilizada nas trocas de informação correspondentes ao processamento dos serviços, e a *Rede de Gestão de Telecomunicação* identificada pelo recurso a uma rede X.25.

Relações funcionais e pontos de referência para o CS-1

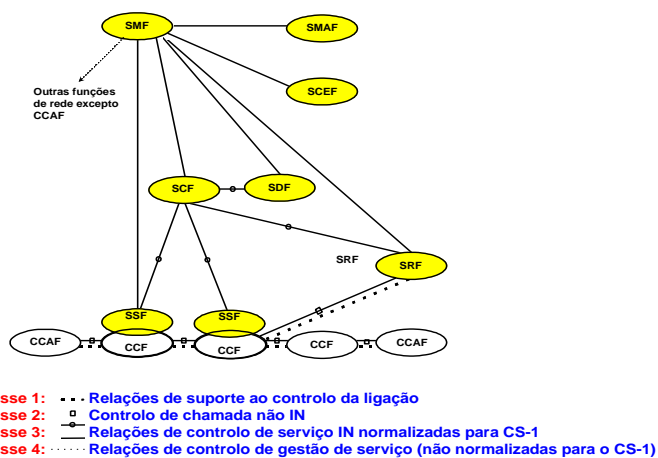


Figura 3 – Relações funcionais e pontos de referência para o CS-1 [1].

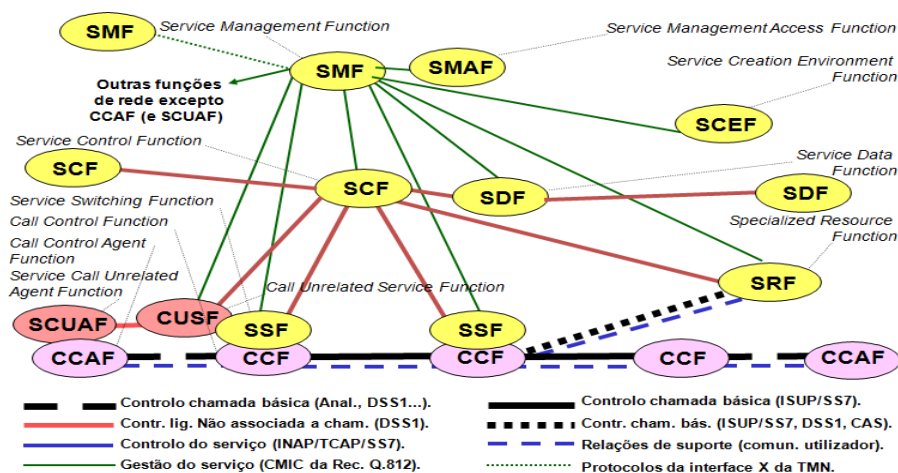


Figura 4 - Relações funcionais e pontos de referência para o CS-2 [9].

Nas Figuras anteriores são apresentadas as relações funcionais e pontos de referência para as entidades funcionais definidas nos primeiros estágios de normalização conhecidos por *ITU-T CS-1 (Capability Set - 1)* e por *ITU-T CS-2 (Capability Set-2)*

2.4 Estratégia para a Introdução da Rede Inteligente

A introdução de serviços *IN* numa rede pública pode trazer ao operador os seguintes benefícios [1, 9]:

- Melhoria do serviço prestado aos seus clientes: novos serviços suplementares, personalização dos serviços;
- Redução de custos, tanto da introdução como da exploração de novos serviços;
- Testes dos serviços efectuados de modo rápido e económico;
- Independência relativamente aos fabricantes dos sistemas para a rede, especialmente de comutação;
- Fixar a sua base de clientes;
- Aumento da rentabilização da infra-estrutura de rede;
- Ciclo de vida dos serviços cada vez mais curtos;
- Optimização de recursos;
- Adaptação às necessidades do cliente;

A introdução de novos serviços nunca é feita num único passo. É necessário ter em consideração os seguintes factores [9]:

- Configuração actual da rede;
- Tipo de serviço a introduzir;
- Tipo de terminais para acesso à Rede;

Temos de encarar também, alguns factores negativos que atrasaram a introdução de novos serviços [9]:

- Normalização tardia (só passado alguns anos após a criação da primeira arquitectura de *IN*, é que surgiram os primeiros organismos de normalização);
- Quase inexistência de produtos conforme as normas (antes de surgirem os primeiros organismos de normalização);
- Receio por parte dos fornecedores de comutadores de perderem o domínio actual sobre os operadores relativo a novos serviços;

2.5 Necessidade de Reconfiguração Automática da RI

Em arquitecturas distribuídas, como é o caso das *Redes Inteligentes*, e onde os utilizadores deste serviço agem de forma independente, poderão acontecer situações em que se exceda a capacidade de carga que os nós e o sistema conseguem processar, podendo causar um colapso total do sistema. É neste sentido que surge a necessidade de *Controlo de Carga* e *Reconfiguração Automática* do sistema. O *Controlo de Carga* aliado à *Reconfiguração Automática* da *RI* vai proporcionar um melhor aproveitamento dos recursos físicos do sistema, tais como por exemplo *CPU* e *RAM*, permitindo também um melhor aproveitamento de recursos humanos. Esta reconfiguração vai servir para controlar a carga de chamadas admitidas pelo sistema garantindo níveis mínimos de qualidade de serviço desejado pelo operador, mesmo na presença de sobrecargas. O *Controlo de Carga* é feito ao nível de rejeição de chamadas no *SCP*.

Esta dissertação analisa vários algoritmos de controlo, dando especial relevo ao controlo *Proporcional* e *Proporcional-Integrador* [7, 8] uma vez que são muito simples de implementar e apresentam baixo *overhead*. No final da análise, são indicadas as vantagens e desvantagens de cada um desses algoritmos de controlo assim como qual será o melhor para o sistema específico que se pretende controlar.

3. Estudo de Algoritmos de Controlo de Carga

3.1 Modelização do Sistema

Foi criado um modelo usando o *MATLAB*, no qual se pretende simular um Sistema de *Redes Inteligentes*, mais propriamente os *SCF* e os *SDF*, e estudar vários Algoritmos de controlo com base no seu desempenho bem como a sua estabilidade em situações de sobrecarga de Rede.

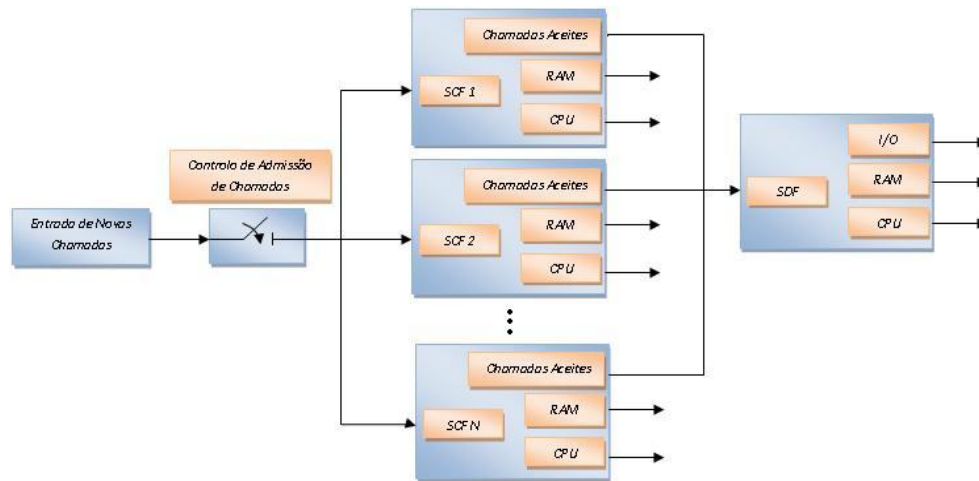


Figura 5 - Modelização do Sistema.

Cada pedido de acesso do utilizador ao *SCF* é feito na forma de Chamada Telefónica (foi escolhido este tipo de serviço, porque é o que consome mais recursos). Cada chamada tem uma duração média de 120 Seg. sendo gerados em média 5 eventos por chamada (um evento no início e no final da chamada e um evento por cada 30 Seg. durante a chamada). O Sistema total tem uma resolução de 0,1 Segundos.

O número de Chamadas activas influencia a capacidade *RAM* do *SCF* (com capacidade máxima para 2000 chamadas activas, sendo ainda acrescentado um “*footprint*” de 10% que está sempre presente mesmo que não haja chamadas activas). Os Eventos gerados por cada chamada influenciam directamente o *CPU* do *SCF* atingindo este a carga máxima instantânea para 365 eventos.

$$RAM_{SCF}(n) = \left(\frac{n^o \text{ Chamadas Activas}(n)}{2000} \cdot 90 \right) + 10 \quad (1)$$

$$CPU_{SCF}(n) = \frac{n^o \text{ Eventos Activos}(n)}{365} \cdot 100 \quad (2)$$

O valor da capacidade máxima de chamadas activas permitidas pelo *SCF* é o mesmo que se encontra definido actualmente no sistema real, sendo o valor máximo de eventos permitidos estimado com base numa aproximação linear de valores de vários *SCF* reais.

SCF	Máximo CPS definido	Nº Actual de CPS	CPU	Nº Estimado de CPS com CPU a 100%
A	15	4,5	6%	75
B	15	7	13%	53,8462
C	15	12	15%	80
D	15	12,4	16%	77,5
E	15	5,5	7%	78,5714
Média				72,9835

Tabela 2 - Valores Reais e Estimados de SCF.

$$N^{\circ} \text{ Estimado de CPS}_{CPU_{100\%}} = \frac{N^{\circ} \text{ Actual CPS} \cdot 100}{CPU} \quad (3)$$

Cada chamada tem em média 120 Segundos gerando assim 5 Eventos, então 15 Chamadas vão gerar 75 Eventos.

$$N^{\circ} \text{ Máximo de Eventos} = \frac{72,9835 \text{ CPS} \times 75 \text{ Eventos}}{15 \text{ Chamadas}} = 364,92 \approx 365 \quad (4)$$

As capacidades de RAM e de CPU do SDF são influenciadas pelos SCF, sendo que o número de Chamadas Activas da cada SCF vai influenciar em 0,1 a capacidade RAM total do SDF, o número de Eventos processados em cada SCF influencia directamente na capacidade total de carga do CPU do SDF e em 0,4 nos estados de transição I/O.

$$RAM_{SDF}(n) = \left(\frac{n^{\circ} \text{ Chamadas Activas}(n) \cdot 0,1}{2000} \cdot 90 \right) + 10 \quad (5)$$

$$CPU_{SDF}(n) = \frac{n^{\circ} \text{ Eventos Activos}(n)}{365} \cdot 100 \quad (6)$$

$$I/O_{SDF}(n) = \frac{n^{\circ} \text{ Eventos Activos}(n) \cdot 0,4}{365} \cdot 100 \quad (7)$$

Cada SCF possui ainda um controlo próprio conforme os Algoritmos de Controlo que pretendem ser estudados, podendo o SDF influenciar nesse controlo.

Foram feitos três tipos de testes, um teste apenas considerando um SCF, um segundo teste considerando um SCF com um SDF agregado e um terceiro tendo considerado 4 SCF, com um esquema de prioridades entre eles e um SDF agregado.

No primeiro teste os limites permitidos para a capacidade RAM e carga no CPU do SCF são respectivamente 70% e 20%.

No Segundo teste os limites impostos no *SCF* foram de 70% para a *RAM* e 20% para o *CPU* e para o *SDF* agregado foram de 15% para a *RAM*, 17% para o *CPU* e 7% para as transições *I/O*.

No terceiro teste os limites impostos em cada *SCF* foram de 70% para a *RAM* e 20% para o *CPU* e para o *SDF* agregado foram de 25% para a *RAM*, 50% para o *CPU* e 20% para as transições *I/O*. O esquema de prioridades é feito em 5 níveis, tendo cada nível a duração de 1 Segundo. No 1º Nível o *SDF* controla os 4 *SCF*, no 2º Nível controla 3 *SCF*, no 3º Nível controla dois *SCF*, no 4º Nível controla apenas um *SCF* e no 5º Nível não controla nenhum dos *SCF*. Pretende-se provar com este teste que o *SDF* consegue controlar o sistema total fazendo um controlo mais apertado nos *SCF* menos prioritários, compensando o facto de não ter controlo sobre os *SCF* mais prioritários durante um determinado período de tempo. Os *SCF* que são controlados pelo *SDF* apresentam pesos diferentes.

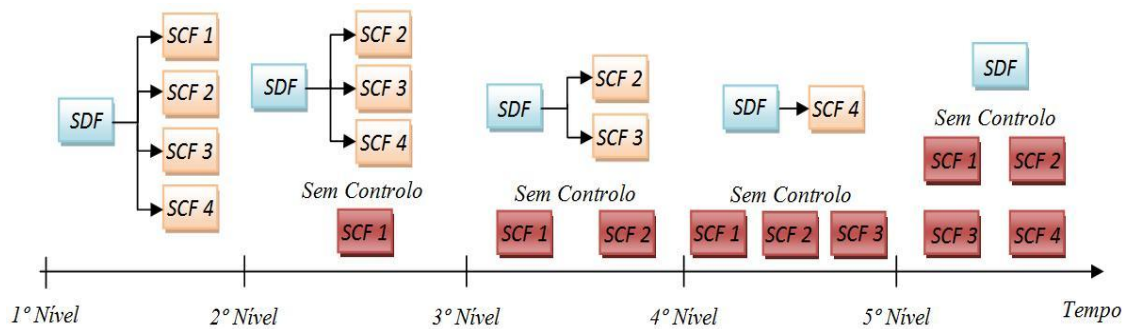


Figura 6 - Esquema do Controlo com quatro SCF.

As principais medidas de comparação serão a carga gerada nos *SCF* e *SDF*, o nível de rejeição de chamadas dos diferentes algoritmos e os erros acumulados de *RAM*, *CPU* e de *I/O*. Este erro pode ser definido como o quadrado do número de vezes em que cada uma destas medidas, *RAM*, *CPU* e *I/O*, ultrapassam os limites impostos, a dividir pelo total de chamadas aceites pelo *SCF*. De notar que o controlo é feito através de rejeição das chamadas e é sempre feito nos *SCF*, podendo os *SDF* influenciar nesse controlo.

$$\text{Erro Acumulado} = EAC = \frac{(N^{\circ} \text{ vezes em que o Limite é ultrapassado})^2}{N^{\circ} \text{ de Chamadas Aceites}} \quad (8)$$

3.2 Geração de Chamadas, Respectiva Duração e Controlo

As chamadas são geradas para que tenham uma taxa de entrada com distribuição de *Poisson* podendo chegar várias chamadas no mesmo instante e tendo o sistema uma resolução de 0,1 Segundos. A duração das chamadas segue uma distribuição *Gaussiana* com média de 120 Segundos. e desvio padrão de 25 Segundos. A criação destas chamadas é feita utilizando dois blocos funcionais da *toolbox communications Blockset* disponíveis no *MATLAB*. Esses blocos funcionais são o *Poisson Integer Generator* e o *Gaussian Noise Generator*.

O *Gaussian Noise Generator* vai criar uma matriz com a duração de cada chamada (Segunda coluna da matriz da *Figura 7*). O *Poisson Integer Generator* vai criar uma matriz de inteiros não negativos, que simbolizam o número de chamadas existentes por instante (Primeira coluna da Matriz da *Figura 7*). Nessa Matriz, um N° inteiro positivo significa que chegou uma chamada nesse instante, se esse N° for precedido de zeros significa que nesse instante chegaram também ao mesmo tempo um N° chamadas iguais ao N° de zeros. O N° 1 significa que haverá chamadas no próximo instante, o N° 2 significa que só haverá chamadas passado um instante, e por ai sucessivamente.

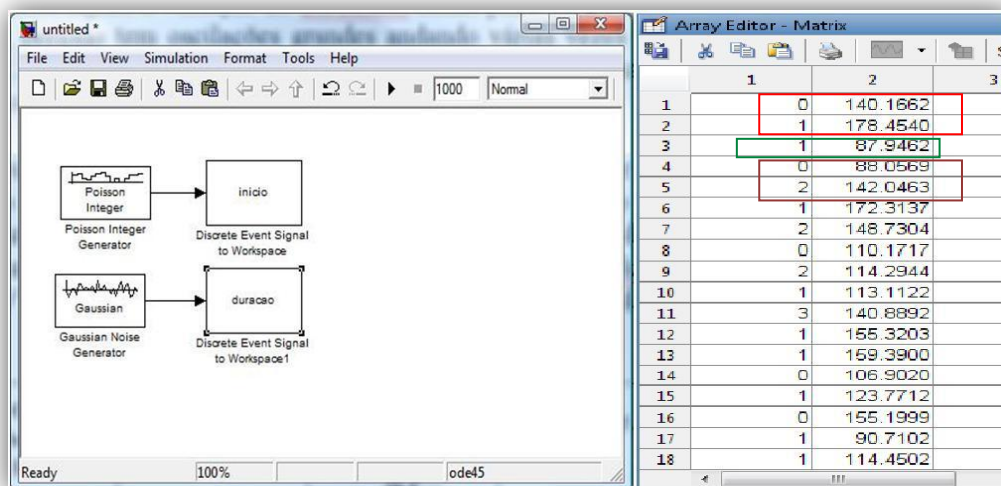


Figura 7 - Geradores de Chamadas e Respetivos Valores.

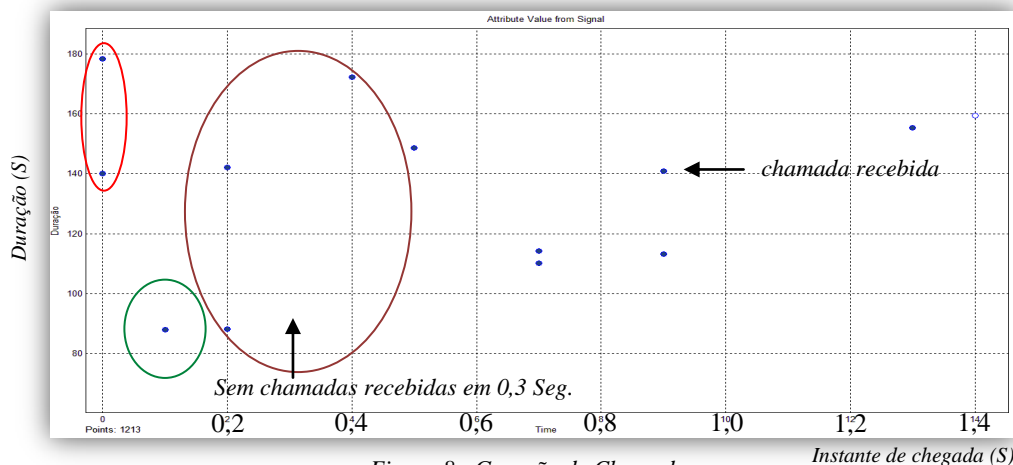


Figura 8 - Geração de Chamadas.

Estas matrizes vão servir como argumentos de entrada de uma função geradora de dados para a simulação. Esta função vai criar os respectivos eventos que cada chamada gera, e vai ter como argumentos de saída o instante de tempo em que cada chamada chegou e a sua respectiva duração, o Nº de Chamadas Activas em cada instante, o Nº de eventos activos em cada instante e o valor da *RAM* e *CPU* do sistema em cada instante.

A rejeição das chamadas pelos algoritmos de controlo é feita colocando a duração das chamadas a zero. Desta forma elas não contam para o Nº de chamadas activas nem geram os seus respectivos eventos.

O padrão de chegada e duração das chamadas é extremamente variável ao longo do dia, de dia para dia, e ao longo do ano. A *Figura 9* mostra o padrão do Nº de chamadas que chegam por hora ao longo de 1 dia no caso de um determinado operador.

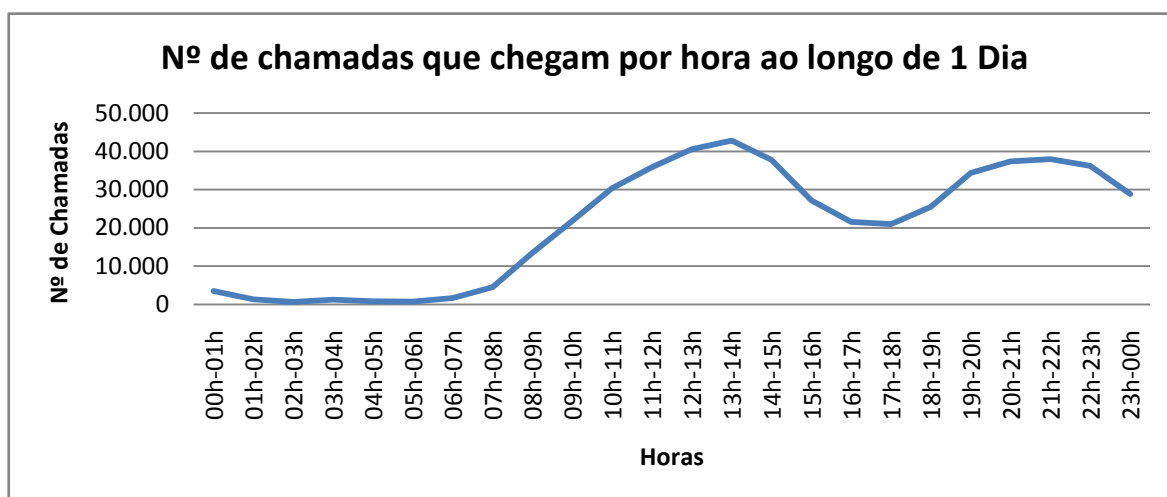


Figura 9 - Nº de chamadas que chegam por hora ao longo de 1 Dia.

Para os estudos que se seguem, em particular para a comparação dos algoritmos de controlo, foi usado um vector de chamadas com duração total de 3600 segundos, e com 4 fases distintas representativas de 4 períodos do dia, cada uma com 900 segundos de duração. O vector de chamadas utilizado está representado na *Figura 10*.

3.3 Simulação sem Controlo

Este ponto apenas serve para demonstrar e observar os dados da simulação de uma Rede Inteligente, mais concretamente de um, ou vários *SCF*, e um *SDF*, que se obtém quando não existe nenhum Algoritmo de Controlo.

3.3.1 Simulação de um SCF

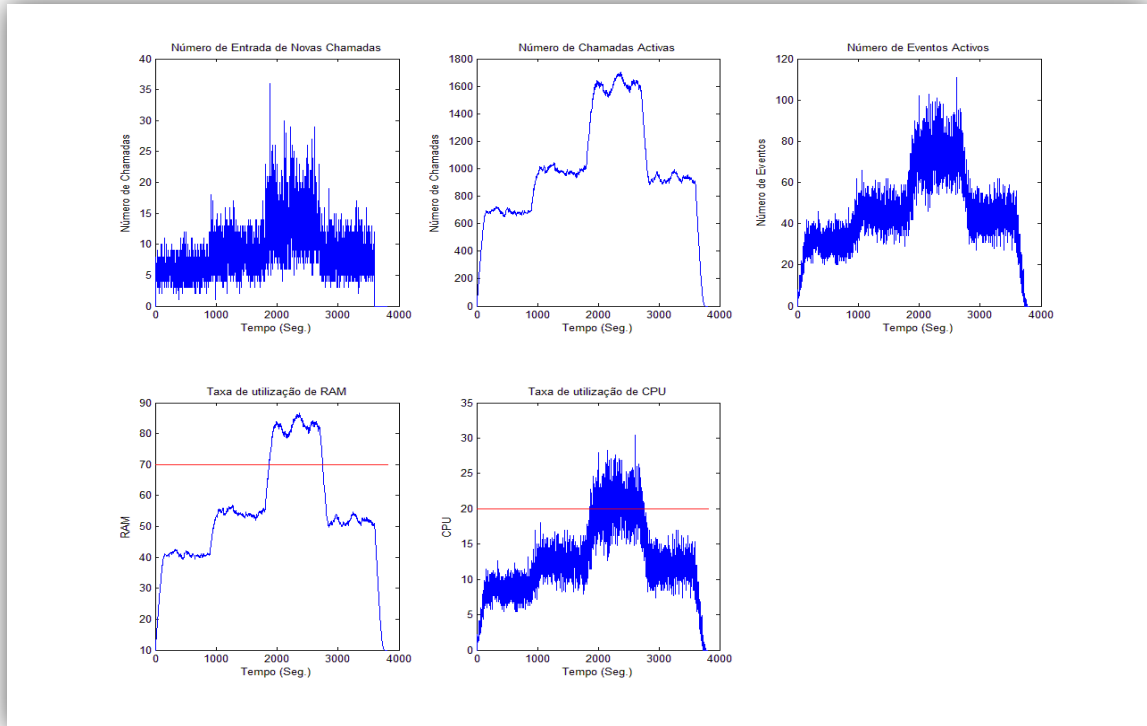


Figura 10 - Simulação sem controlo de um SCF.

Os limites impostos ao SCF estão sinalizados com uma linha vermelha. Podemos verificar que esses limites foram ultrapassados. Podemos verificar também que o *Número de Chamadas Activas* é semelhante à *Taxa de utilização de RAM* e o *Número de Eventos Activos* é semelhante à *Taxa de Utilização de CPU* e por isso, a partir deste ponto não se irão representar mais os gráficos do *Número de Chamadas Activas* e os gráficos do *Número de Eventos Activos*. Esta semelhança deve-se à dependência directa da RAM com o número da Chamadas Activas e do CPU com o número de Eventos Activos como podemos confirmar pelas fórmulas (1) e (2).

3.3.2 Simulação de um SCF e um SDF

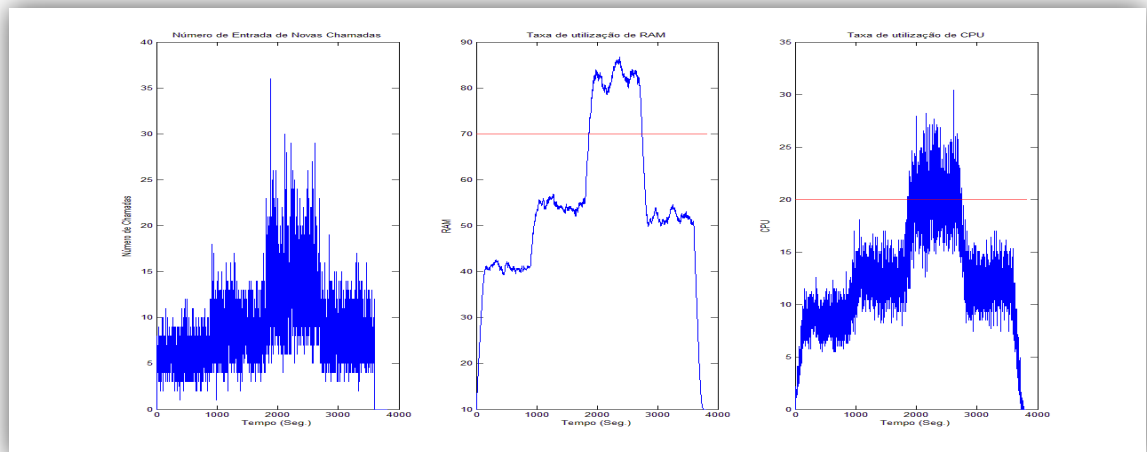


Figura 11 - Simulação sem controlo do SCF.

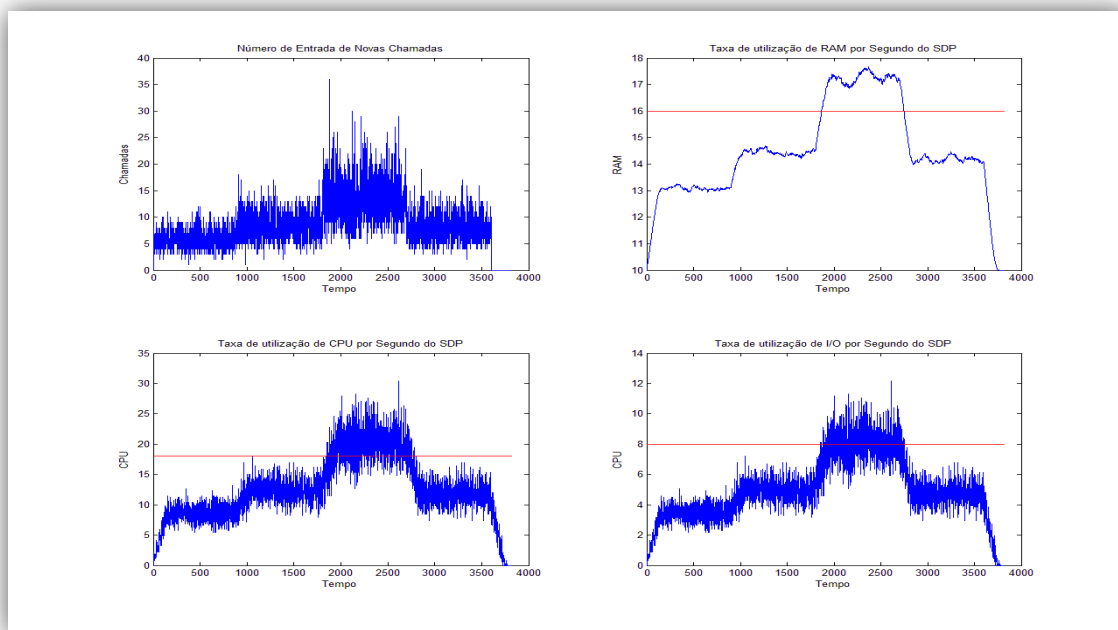


Figura 12 - Simulação sem controlo do SDF.

Os limites impostos ao *SCF* e ao *SDP* estão sinalizados com uma linha vermelha. Podemos verificar que também neste caso esses limites foram ultrapassados.

3.3.3 Simulação de quatro *SCF* e um *SDF*

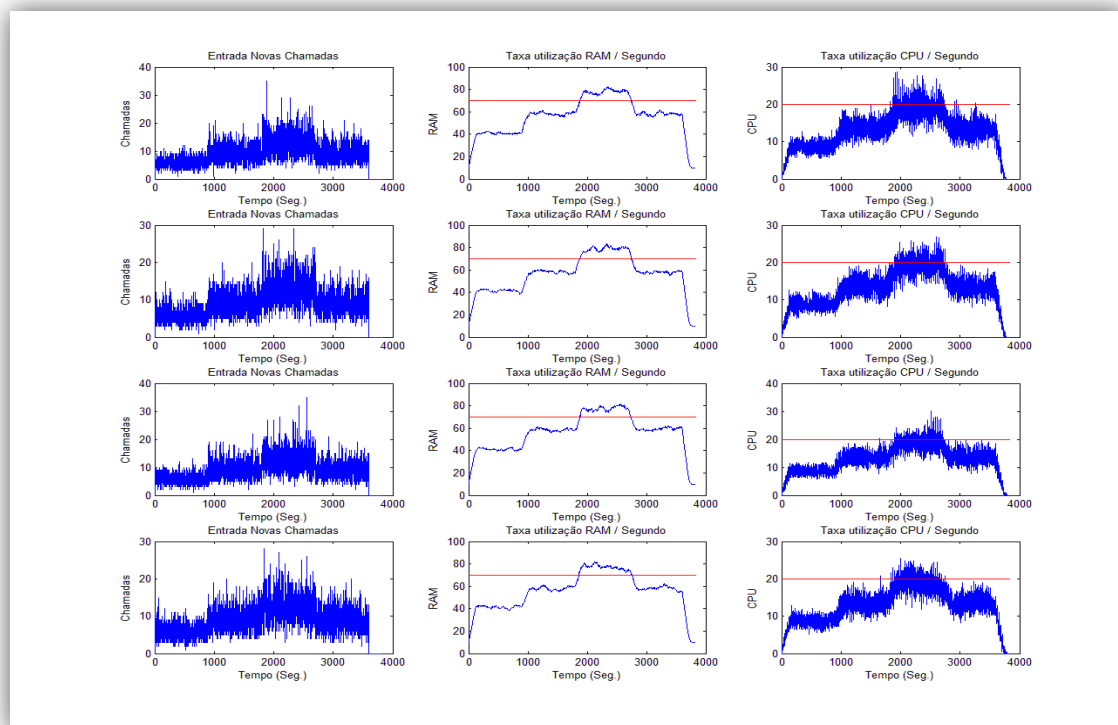


Figura 13 - Simulação sem controlo dos quatro SCF.

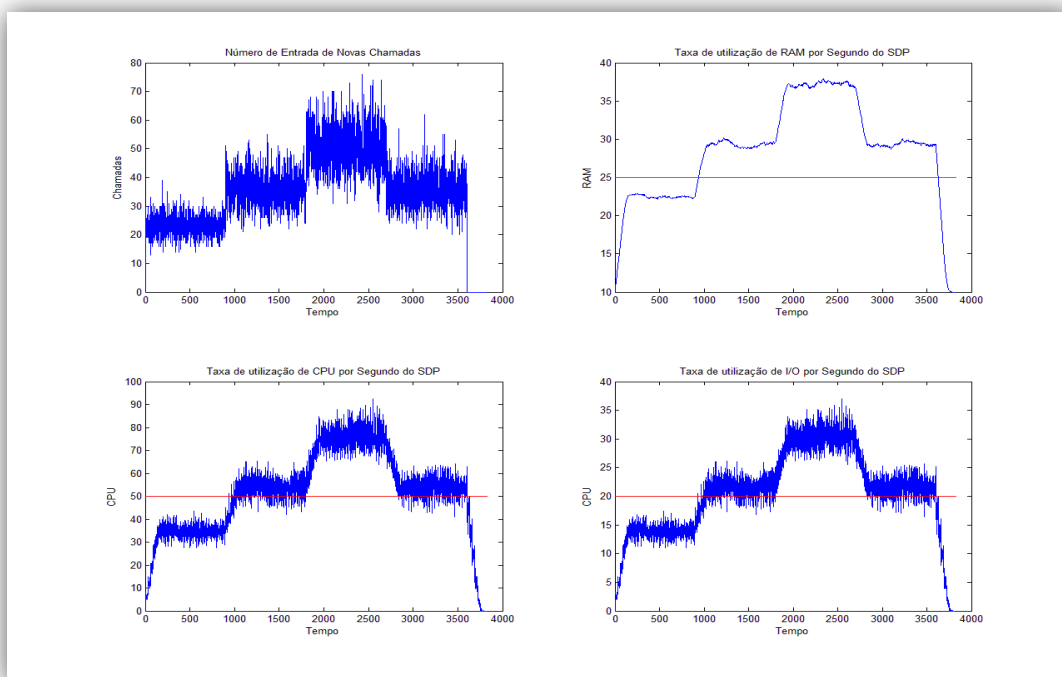


Figura 14 - Simulação sem controlo do SDF com influência dos quatro SCF.

Os limites impostos aos 4 SCF e ao SDF estão sinalizados com uma linha vermelha. Podemos verificar esses limites foram ultrapassados em todos os SCF e no SDF.

3.4 Simulação do Controlo com um *Limiar*

O funcionamento deste controlo é bastante simples e funciona fixando um *Limiar*, isto é, um limite máximo para o número de entrada de novas chamadas no SCF. Sempre que o *Limiar* for ultrapassado num determinado instante, o SCF não aceita mais chamadas novas deste o instante seguinte até ao final do Segundo correspondente desse instante (não esquecer que a resolução do sistema é de 0,1 Segundos, o que faz com que em cada Segundo tenhamos 10 instantes). Isto implica que mesmo que o *Limiar* seja '0' há sempre aceitação de chamadas no primeiro instante em que chegam chamadas, uma vez que para que o controlo de rejeição de chamadas seja activado é necessário que o *Limiar* seja ultrapassado. De notar que a rejeição de chamadas implica necessariamente uma diminuição da taxa de ocupação de RAM e CPU devido a não se permitirem que entrem novas chamadas no sistema para substituir as chamadas activas que estão em fim de vida.



Figura 15 - Esquema de Aceitação e Rejeição do SCF.

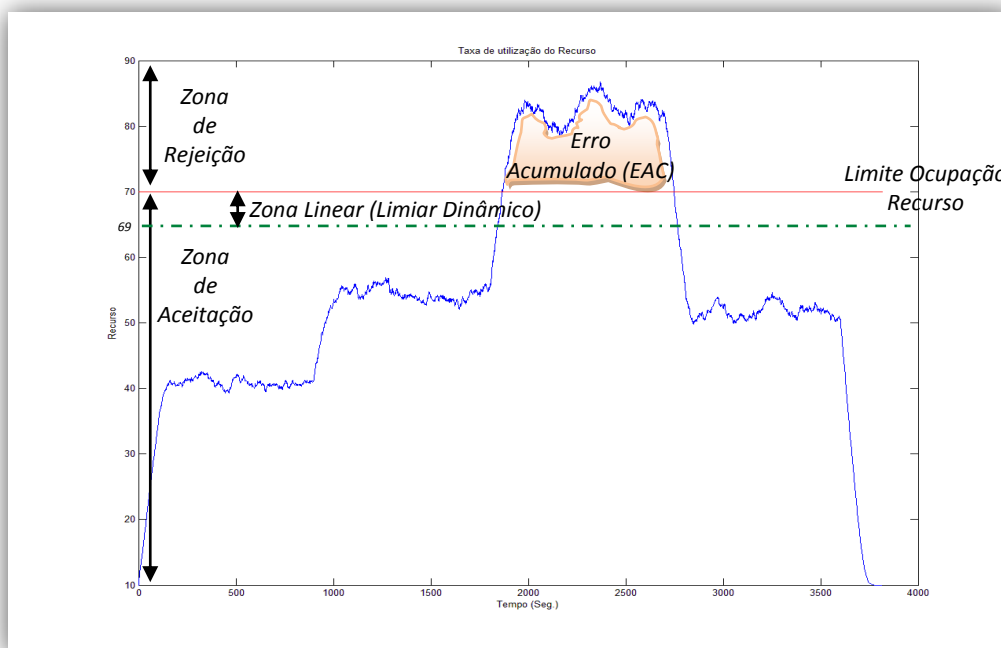


Figura 16 - Controlo com um Limiar.

3.4.1 Controlo de um SCF

i) Um Limiar Estático

Este controlo é o que se encontra implementado de momento no sistema de Redes que se pretende melhorar. Foi feita a simulação deste controlo para podermos comparar com os outros Algoritmos de controlo que pretendemos estudar.

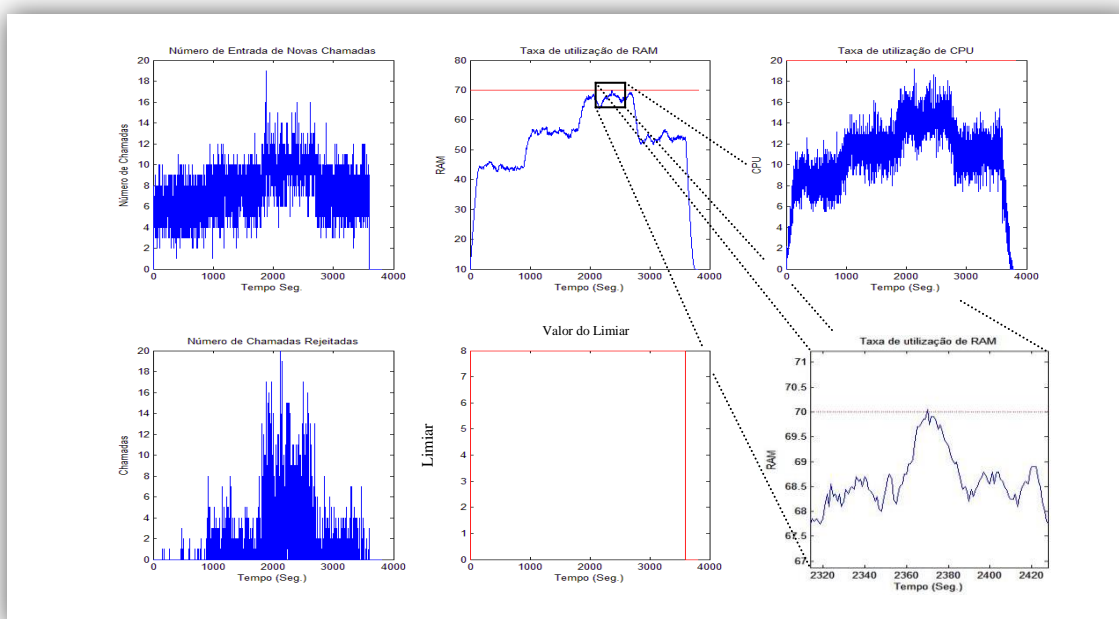


Figura 17 - Simulação do controlo do SCF com um Limiar Estático.

Este tipo de controlo possui várias limitações. Uma dessas limitações é o facto de não aproveitar eficientemente os recursos físicos dos *SCF* e dos *SDF*, e de desperdiçar recursos humanos na manutenção da Rede, isto é, como o controlo do *Limiar* não é feito de forma automática é necessário que alguém diminua ou aumente esse *Limiar* para que o *SCF* ou o *SDP* cumpram com os limites pretendidos.

Nesta simulação foi necessário ir baixando o *Limiar* manualmente para podermos cumprir com os limites impostos ao *SCF* e *SDF*.

Valor <i>Limiar</i> Estático (CPS)	Nº Chamadas Chegadas	Nº Chamadas Rejeitadas	Percentagem Chamadas Rejeitadas	Erro Acumulado RAM SCF	Erro Acumulado CPU SCF
8	31 506	4 316	13,6990 %	$1,0872 \times 10^{-5}$	0

Tabela 3 - Resultado da Simulação do controlo do SCF com um *Limiar* Estático.

Podemos afirmar que o controlo foi conseguido, estando os valores da ocupação de *RAM* e de *CPU* dentro dos limites pretendidos. O valor limite pretendido para a *RAM* é ultrapassado ligeiramente aos 2370 Seg. verificando-se por isso *Erro Acumulado de RAM*.

ii) Um *Limiar* Dinâmico com Controlo Proporcional

Este controlo é feito fazendo variar dinamicamente o *Limiar* conforme as medidas de ocupação de *RAM* e carga do *CPU* do *SCF*. Estas medidas são actualizadas em cada instante, isto é, em cada 0,1 Segundos, e a partir delas e através de um controlo *Proporcional* obtêm-se a seguinte malha de controlo que vai permitir controlar o sistema em cada instante:

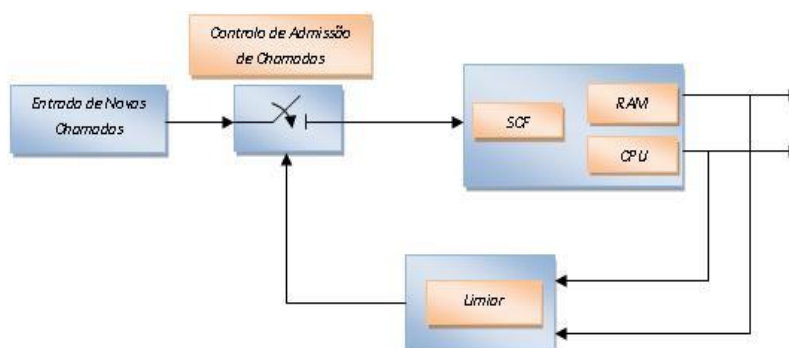


Figura 18 - Malha de Controlo do SCF com *Limiar* Dinâmico

O *Limiar Dinâmico* é calculado usando o método do *Controlo Proporcional*, tendo as fórmulas (9) e (10). Note-se que $x|_0^1$ significa que o valor de x é truncado a 0 e 1 se for menor que 0 ou maior que 1 respectivamente, $x|_0^1 = \begin{cases} 1, x \geq 1 \\ x, 0 < x < 1, x \in \mathbb{R} \\ 0, x \leq 0 \end{cases}$.

$$eP_{SCF(n)} = \left(k_p \cdot (\gamma^T - \gamma(n)) \right) \Big|_0^1 \cdot \left(\beta^T - \beta(n) \right) \Big|_0^1 \Big|_0^1 \quad (9)$$

$$Limiar(n) = eP_{SCF(n)} \cdot Limiar_{Máximo} \quad (10)$$

K_p : Ganho Proporcional do Controlador

γ^T : Limite Pretendido de Ocupação de RAM

γ : Valor Actual de Ocupação de RAM

β^T : Limite Pretendido de carga no CPU

β : Valor Actual de Carga no CPU

eP_{SCF} : erro combinado de RAM e CPU

O *Limiar* calculado com este método vai actuar no controlo de admissão de chamadas de forma semelhante ao método do *Limiar Estático* descrito no ponto anterior, com a diferença de que este *Limiar* varia automaticamente conforme a taxa de ocupação de RAM e carga no CPU do SCF, como já foi referido em cima, sem ser necessário um controlo manual e adaptando-se ao sistema que pretendemos controlar aproveitando melhor os recursos do sistema.

Podemos observar através da *Figura 19* que a ocupação de RAM do SCF no período mais crítico se encontra dentro dos limites de controlo. Podemos observar ainda a variação do *Limiar* quando a taxa de ocupação de RAM e CPU se aproximam dos limites impostos, mais concretamente, dentro da zona linear do controlo do *Limiar Dinâmico*. O *Limiar* nesta zona tem oscilações grandes andando várias vezes entre o valor máximo e o '0'. Este facto deve-se ao fácil efeito de saturação da equação devido a termos um ganho proporcional k_p elevado.

Valor <i>Limiar</i> Inicial (CPS)	Valor K_p	Nº Chamadas Chegadas	Nº Chamadas Rejeitadas	Percentagem Chamadas Rejeitadas	Erro Acumulado RAM SCF	Erro Acumulado CPU SCF
20	3	31 506	2032	6,4496 %	$1,9287 \times 10^{-5}$	0,0020

Tabela 4 - Resultados da Simulação do Controlo do SCF com um *Limiar Dinâmico Proporcional*.

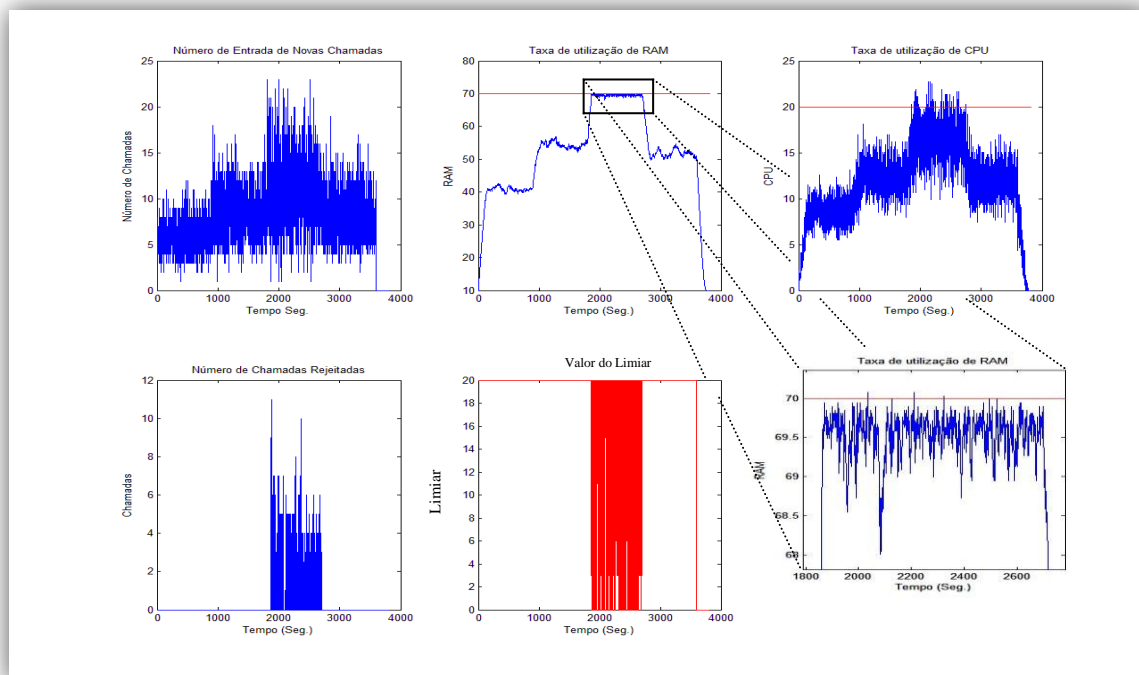


Figura 19 - Simulação do Controlo do SCF com um Limiar Dinâmico Proporcional.

O limite desejado de carga de *CPU* do *SCF* foi ultrapassado numa margem considerável entre os 1800 e os 2800 Segundos. Isto acontece porque o *CPU* depende dos eventos gerados pelas chamadas, e visto que as chamadas geram eventos durante toda a duração destas, influenciam assim o valor do *CPU* do *SCF* no presente e no futuro, tornando assim o controlo do *CPU* mais difícil. É também devido a isto que temos um erro acumulado de *CPU* bem maior do que o erro acumulado de *RAM*, uma vez que o seu respectivo limite é muito mais vezes violado. Foram aceites mais 2 300 chamadas que no caso anterior, o que faz com que a carga de *CPU* suba consideravelmente. Este facto obriga a que o *Limiar* venha várias vezes a '0' contribuindo assim consideravelmente para a rejeição de chamadas.

Os K_p foram afinados com valores compreendidos entre '0' e '100' sendo escolhido os valores que apresentavam a menor taxa de rejeição de chamadas. Durante esta afinação não foi observada qualquer situação de instabilidade.

iii) Um *Limiar* Dinâmico com Controlo Proporcional-Integrador

Este controlo actua de forma semelhante ao controlo *Proporcional* no cálculo dinâmico do *Limiar*, com a diferença de que neste caso é utilizado o controlo *Proporcional-Integrador* para esse cálculo, sendo a malha de controlo semelhante, mas com fórmulas diferentes.

$$eT_{SCF}(n) = \left((Y^T - Y(n)) \Big|_0^1 \cdot (\beta^T - \beta(n)) \Big|_0^1 \right) \quad (11)$$

$$eI_{SCF}(n) = \sum_{i=1}^{N=10} eT_{SCF}(n-i) \quad (12)$$

$$ePI_{SCF}(n) = \left((k_p \cdot eT_{SCF}(n)) + (k_i \cdot eI_{SCF}(n)) \right) \Big|_0^1 \quad (13)$$

$$Limiar(n) = ePI_{SCF}(n) \cdot Limiar_{Máximo} \quad (14)$$

eI_{SCF} : Factor Integrador do SCF

ePI_{SCF} : Factor Proporcional integrador

k_p : Ganho Proporcional do Controlador

k_i : Ganho Integrador do Controlador

eT_{SCF} : erro combinado de RAM e CPU

Observamos na *Figura 20* que com um controlo *Proporcional-Integrador* a taxa de ocupação de RAM do SCF no período mais crítico ultrapassa o valor limite imposto tendo uma variação média de $\pm 0,5$. Podemos dizer então que existe controlo mas o controlo do valor limite é pior com este controlador (típico de um controlador *PI*, devendo-se isto ao *Integrador* [7]). O lado positivo é que devido a este facto o valor da rejeição de chamadas diminui uma vez que foram aceites mais chamadas. Continuamos a observar grandes oscilações do *Limiar* entre o valor máximo e '0', facto este que já foi referido anteriormente e que se deve ao efeito de saturação de equação de controlo devido a termos ganhos K_p e K_i elevados.

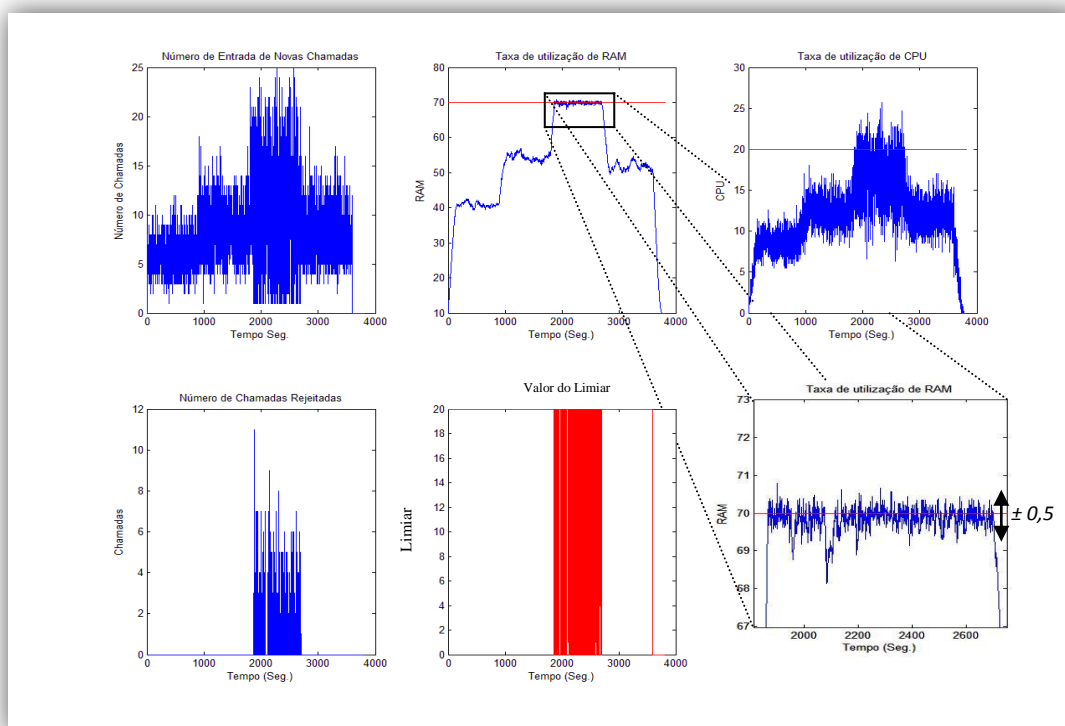


Figura 20 - Simulação do Controlo do SCF com um Limiar Dinâmico PI.

O limite desejado de carga de *CPU* do *SCF* também continua a ser ultrapassado com este controlo devido ao efeito já explicado anteriormente, e como também foi explicado, este facto contribui também para a rejeição das chamadas. Também aqui os valores de K_p e K_i foram afinados com valores compreendidos entre '0' e '100' sendo escolhido os valores que apresentavam a menor taxa de rejeição de chamadas. Durante esta afinação não foi observada qualquer situação de instabilidade.

Valor Limiar Inicial (CPS)	Valor K_p	Valor K_i	Nº Chamadas Chegadas	Nº Chamadas Rejeitadas	Percentagem Chamadas Rejeitadas	Erro Acumulado RAM SCF	Erro Acumulado CPU SCF
20	2	4,5	31 506	1 993	6,3258 %	0,0040	0,0118

Tabela 5- Resultados da Simulação do Controlo do SCF com um Limiar Dinâmico PI.

3.4.2 Controlo de um SCF e um SDF

Como já foi referido o controlo de rejeição de chamadas é sempre feito no *SCF* podendo este controlo ser influenciado também pelo *SDF*. Além das técnicas de controlo no *SCF* explicadas nos pontos anteriores o controlo no *SDF* também pode ser do tipo *Estático*, *Proporcional*, ou *Proporcional-Integrador*. As medidas de ocupação de *RAM*, carga do *CPU* e *I/O* do *SDF* são também actualizadas em cada instante, o que permite que também neste caso se faça um controlo ao instante.

Temos então a malha de controlo do sistema total para o caso do controlo *Proporcional* e *Proporcional-Integrador*:

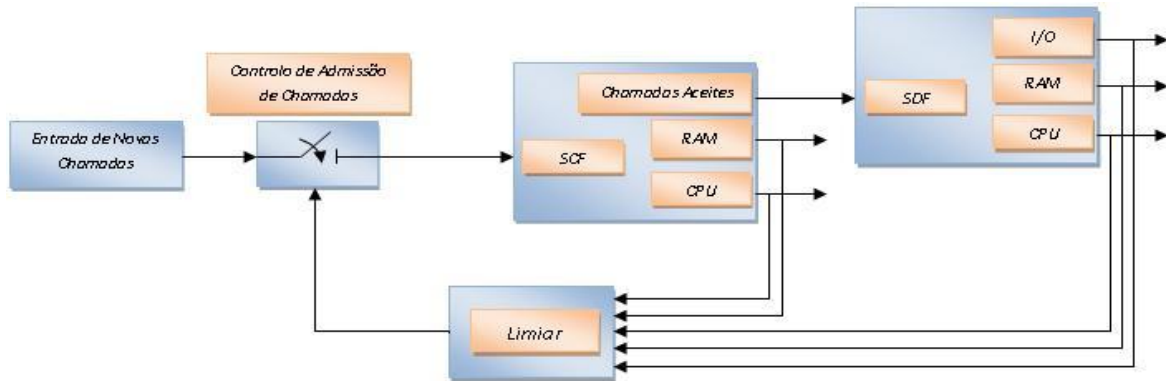


Figura 21 – Malha de Controlo de um SCF e de um SDF com Limiar Dinâmico.

No caso do *Controlo Proporcional* o *Limiar Dinâmico* é calculado com as fórmulas [15] e [16]:

$$eP_{SDF}(n) = \left(k_{p_{SDF}} \cdot (Y^T - Y(n)) \Big|_0^1 \cdot (\beta^T - \beta(n)) \Big|_0^1 \cdot (\lambda^T - \lambda(n)) \Big|_0^1 \right) \quad (15)$$

$$Limiar(n) = Limiar_{Máximo} \cdot Control_{SCF}(n) \cdot eP_{SDF}(n) \quad (16)$$

λ^T : Limite Pretendido de carga I/O do SDF

λ : Valor Actual de Carga no I/O do SDF

$Control_{SCF}$: Factor de Controlo do SCF (pode ser o valor do controlo *Proporcional*, *Proporcional-Integrador*, obtidos com as fórmulas (9) e (13) respectivamente, ou o pode ter o valor 1 se o controlo do SCF for estático)

No caso do *Controlo Proporcional-Integrador* o *Limiar Dinâmico* é calculado com as fórmulas (17) a (20):

$$eT_{SDF}(n) = (Y^T - Y(n)) \Big|_0^1 \cdot (\beta^T - \beta(n)) \Big|_0^1 \cdot (\lambda^T - \lambda(n)) \Big|_0^1 \quad (17)$$

$$eI_{SDF}(n) = \sum_{i=1}^{N=10} eT_{SDF}(n-i) \quad (18)$$

$$ePI_{SDF}(n) = \left(\left(k_{P_{SDF}} \cdot eT_{SDF}(n) \right) + \left(k_{I_{SDF}} \cdot eI_{SDF}(n) \right) \right) \Big|_0^1 \quad (19)$$

$$Limiar(n) = Limiar_{Máximo} \cdot Control_{SCF}(n) \cdot ePI_{SDF}(n) \quad (20)$$

Em baixo podemos observar uma tabela com os resultados dos diversos controlos do SCF e SDF. O *Limiar*, o K_p e K_I do SCF já foram apresentados no ponto 3.4.1.

	$K_{P_{SDF}}$	$K_{I_{SDF}}$	Nº Chamadas Chegadas	Nº Chamadas Rejeitadas	% Chamadas Rejeitadas	EAC RAM SCF	EAC CPU SCF	EAC RAM SDF	EAC CPU SDF	EAC I/O SDF
Controlo SCF: Estático Controlo SDF: Proporcional	105	-----	31 506	4 691	14,8892 %	0	0	$1,6852 \times 10^{-7}$	$4,7948 \times 10^{-4}$	$4,1167 \times 10^{-5}$
Controlo SCF: Estático Controlo SDF: PI	35	8	31 506	4 405	14,8321 %	0	0	$2,0506 \times 10^{-7}$	$5,2373 \times 10^{-4}$	$4,3852 \times 10^{-5}$
Controlo SCF: Proporcional Controlo SDF: Proporcional	35	-----	31 506	3 385	11,7692 %	0	$3,9329 \times 10^{-6}$	$8,2269 \times 10^{-7}$	0,0026	$1,9897 \times 10^{-4}$
Controlo SCF: Proporcional Controlo SDF: PI	25	4	31 506	3 297	11,6391 %	0	$3,5789 \times 10^{-4}$	$2,4546 \times 10^{-5}$	0,0105	$9,9992 \times 10^{-4}$
Controlo SCF: PI Controlo SDF: Proporcional	65	-----	31 506	3 715	11,7914 %	0	$6,8825 \times 10^{-5}$	$9,7885 \times 10^{-7}$	0,0051	$4,3353 \times 10^{-4}$
Controlo SCF: PI Controlo SDF: PI	45	5	31 506	3 661	11,6200 %	0	$2,6547 \times 10^{-4}$	$3,6025 \times 10^{-5}$	0,0159	0,0015

Tabela 6- Resultados do Controlo do SCF e SDF (um Limiar).

Observando a *Tabela 6* podemos verificar que com o controlo *PI* no SCF e no SDF temos a menor rejeição de chamadas de todos os testes efectuados. Por outro lado, esta combinação também tem os maiores valores de erro acumulado. Isto deve-se às próprias características do controlador e do mecanismo de rejeição de chamadas como já foi explicado.

Apesar disto, podemos dizer que o sistema se encontra controlado uma vez que se encontra numa vizinhança apertada dos limites impostos tendo um erro de $\pm 0,05$ no caso da RAM no SDF. No CPU verifica-se uma violação grande dos limites impostos devido a razões já explicadas anteriormente.

De notar que a combinação de controlo *PI* no SCF e controlo *Proporcional* no SDF aumenta a rejeição de chamadas.

Os K_p e k_i foram afinados com valores compreendidos entre '0' e '100' sendo escolhido os valores que apresentavam a menor taxa de rejeição de chamadas. Durante esta afinação não foi observada qualquer situação de instabilidade.

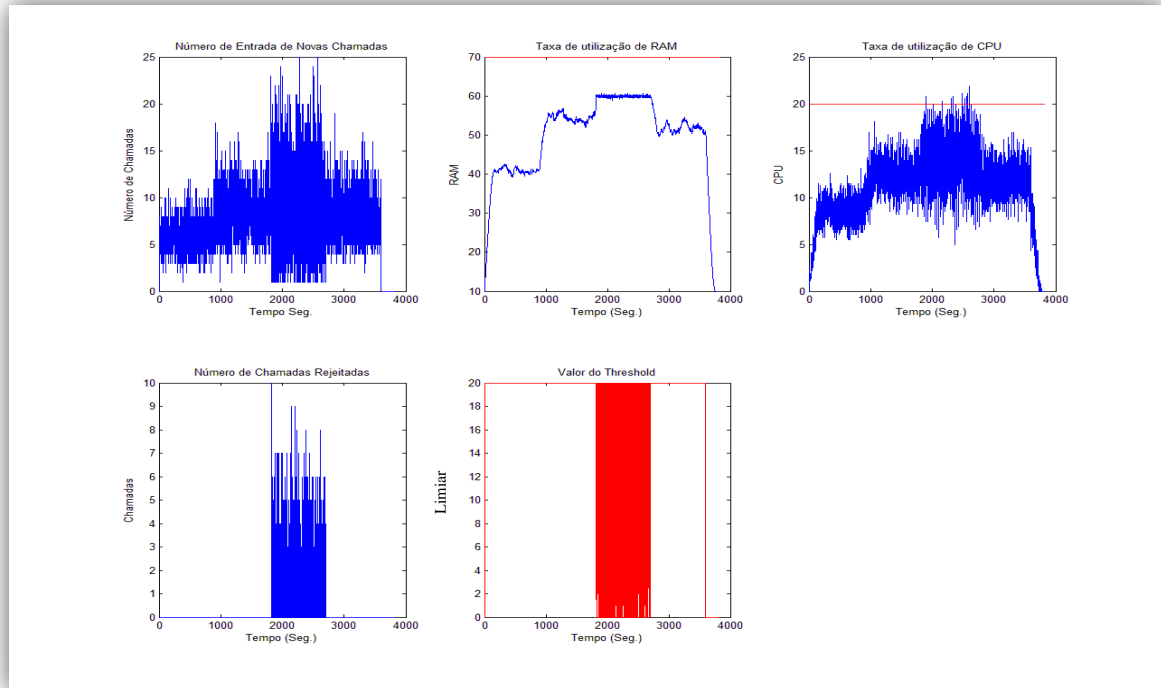


Figura 22 – SCF: Simulação com Controle PI no SCF e SDF (melhor resultado).

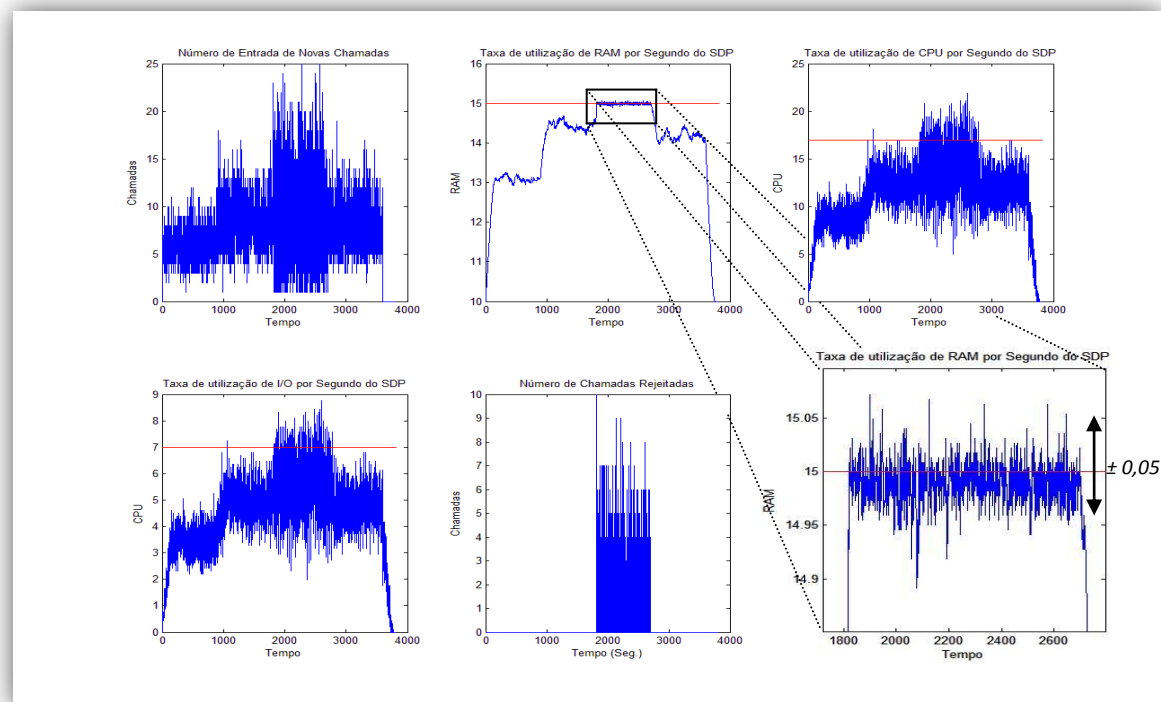


Figura 23 - SDF: Simulação com controle PI no SCF e SDF (melhor resultado).

3.4.3 Controlo de quatro SCF e um SDF

Como já foi explicado, nesta simulação existem quatro SCF com prioridades diferentes. O esquema de prioridades é feito em 5 níveis, tendo cada nível a duração de 1 Segundo. No 1º Nível o SDF controla os quatro SCF, no 2º Nível controla três SCF, no 3º Nível controla dois SCF, no 4º Nível controla apenas um SCF, não controlando nenhum dos SCF no 5º Nível (ver Figura 6). Como já foi dito também pretende-se provar com este teste que o SDF consegue controlar o sistema total fazendo um controlo mais apertado nos SCF menos prioritários, compensando o facto de não ter controlo sobre os SCF mais prioritários durante um determinado período de tempo. Os SCF que são controlados pelo SDF apresentam pesos diferentes. Visto que quando for escolhido um algoritmo de controlo para implementação prática, todos os SCF terão esse algoritmo, também aqui na simulação os SCF terão sempre o mesmo tipo de controlo, isto é, ou terão todos controlo Estáticos, ou Proporcional, ou Proporcional-Integrador.

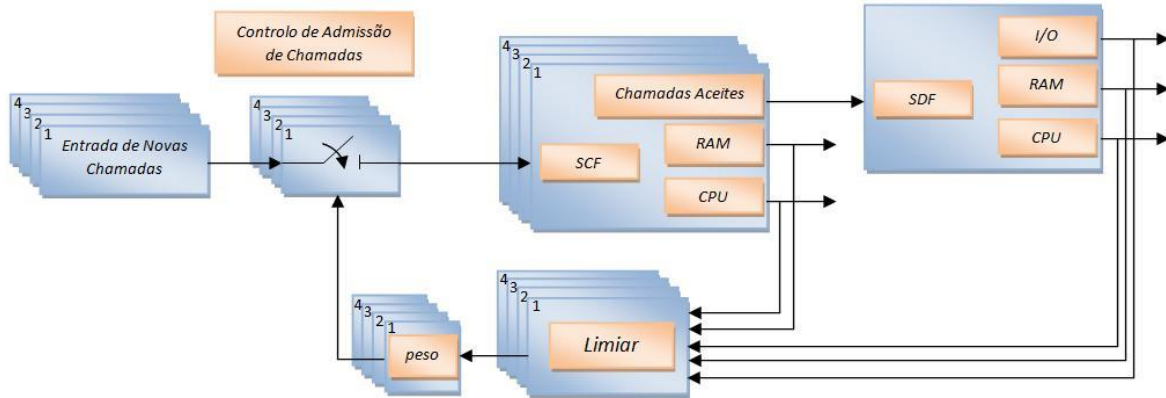


Figura 24 - Malha de Controlo de quatro SCF e de um SDF com Limiar Dinâmico.

A Figura 24 representa a malha de controlo do sistema sendo as fórmulas para esse controlo as seguintes:

Controlo Proporcional:

$$Limiar_{SCF_{ij}}(n) = \begin{cases} peso_{ij} \cdot Limiar_{Máximo} \cdot Control_{SCF_{ij}}(n) \cdot eP_{SDF}(n), & peso_{ij} \neq 0 \\ Limiar_{Máximo} \cdot Control_{SCF_{ij}}(n), & peso_{ij} = 0 \end{cases} \quad (21)$$

i representa o número do SCF e j representa o nível em que esse SCF se encontra. Note-se que quando $peso_{ij} = 0$ não existe influencia do SDF nos SCF.

Controlo Proporcional-Integrador:

$$Limiar_{SCF_{ij}}(n) = \begin{cases} peso_{ij} \cdot Limiar_{Máximo} \cdot Control_{SCF_{ij}}(n) \cdot ePI_{SDF}(n), & peso_{ij} \neq 0 \\ Limiar_{Máximo} \cdot Control_{SCF_{ij}}(n), & peso_{ij} = 0 \end{cases} \quad (22)$$

		peso _{i,j}	SCF			
			1	2	3	5
Nível	1	1/2	1/4	1/8	1/8	
	2	0	1/2	1/4	1/4	
	3	0	0	1/2	1/2	
	4	0	0	0	1	
	5	0	0	0	0	

Tabela 7 - Pesos dos SCF em cada nível.

Para o cálculo do ePI_{SDF} existe uma ligeira alteração na fórmula (18), em vez de utilizar um somatório de 10 amostras utiliza um de 50 amostras. Esta alteração deve-se ao facto de termos uma actuação do algoritmo de controlo durante 5 segundos, sendo necessário utilizar os valores das amostras desses segundos no *Integrador*.

$$eI_{SDF}(n) = \sum_{i=1}^{N=50} eT_{SDF}(n-i) \quad (23)$$

Em baixo podemos observar uma tabela com os resultados dos diversos controlos dos 4 SCF.

	SCF	Limiar Inicial	K_p	K_I	Nº Chamadas Chegadas	Nº Chamadas Rejeitadas	% Chamadas Rejeitadas	EAC RAM SCF	EAC CPU SCF
Controlo SCF: Estático	1º	8	----	---	32 505	5 813	17,8834 %	0	0
	2º	8	----	---	32 750	7 936	24,2321 %	0	0
	3º	8	----	---	32 757	10 698	32,6587 %	0	0
	4º	8	----	---	32 488	10 891	33,5231 %	0	0
Controlo SCF: Estático	1º	8	----	---	32 505	6 085	18,7202 %	0	0
	2º	8	----	---	32 750	8 162	24,9221 %	0	0
	3º	8	----	---	32 757	10 059	30,7079 %	0	0
	4º	8	----	---	32 488	10 391	31,9841 %	0	0
Controlo SCF: Proporcional	1º	20	2,5	---	32 505	2 656	8,1711 %	0	0,0018
	2º	20	2,5	---	32 750	5 527	16,8763 %	0	$1,2747 \times 10^{-4}$
	3º	20	2,5	---	32 757	10 967	33,4739 %	0	0
	4º	20	2,5	---	32 488	15 499	47,7068 %	0	0
Controlo SCF: Proporcional	1º	20	2,5	---	32 505	3 517	10,8199 %	0	0,0033
	2º	20	2,5	---	32 750	7 155	21,8473 %	0	$9,0603 \times 10^{-4}$
	3º	20	2,5	---	32 757	10 592	32,3351 %	0	0
	4º	20	2,5	---	32 488	12 752	39,2514 %	0	0
Controlo SCF: PI	1º	20	2	4,5	32 505	2 571	7,9096 %	0	0,0031
	2º	20	2	4,5	32 750	5 628	17,1847 %	0	$1,5885 \times 10^{-4}$
	3º	20	2	4,5	32 757	10 903	33,2845 %	0	0
	4º	20	2	4,5	32 488	15 520	47,7715 %	0	0
Controlo SCF: PI	1º	20	2	4,5	32 505	3 439	10,5799 %	0	0,0056
	2º	20	2	4,5	32 750	7 375	22,5191 %	0	$9,1387 \times 10^{-4}$
	3º	20	2	4,5	32 757	9 752	29,7707 %	0	$9,0603 \times 10^{-4}$
	4º	20	2	4,5	32 488	13 438	41,3630 %	0	0

Tabela 8 – SCF: Resultados da Simulação com quatro SCF e um SDF.

	K_{p_SDF}	K_{I_SDF}	Nº Chamadas Chegadas	Nº Chamadas Rejeitadas	% Chamadas Rejeitadas	EAC RAM SDF	EAC CPU SDF	EAC I/O SDF
Controlo SCF: Estático Controlo SDF: Proporcional	80	-----	130 500	35 338	27,0789 %	$1,1304 \times 10^{-5}$	0,0077	0,0012
Controlo SCF: Estático Controlo SDF: PI	25	85	130 500	34 697	26,5877 %	0,0045	0,0264	0,0042
Controlo SCF: Proporcional Controlo SDF: Proporcional	70	-----	130 500	34 649	26,5510 %	$6,1411 \times 10^{-4}$	0,0581	0,0093
Controlo SCF: Proporcional Controlo SDF: PI	70	80	130 500	34 016	26,0659 %	0,0205	0,4027	0,0644
Controlo SCF: PI Controlo SDF: Proporcional	80	-----	130 500	34 622	26,5303 %	$6,3106 \times 10^{-4}$	0,0645	0,0103
Controlo SCF: PI Controlo SDF: PI	15	40	130 500	34 004	26,0567 %	0,0211	0,3374	0,0540

Tabela 9 - SDF: Resultados da Simulação com quatro SCF e um SDF.

O resultado com menor rejeição de chamadas é obtido com a combinação controlo *PI* nos *SCF* e no *SDP*. Mais uma vez podemos constatar que os controladores *PI* aumentam os erros acumulados de *RAM*, *CPU* e *I/O* devendo-se isso às próprias características do controlador que faz com que se aceitem mais chamadas.

Podemos dizer ainda que se pode considerar o sistema controlado, andando os valores de ocupação de *RAM* numa vizinhança de $\pm 0,5$ do valor limite estipulado. Os Valores de *CPU* continuam a sofrer grandes variações ultrapassando consideravelmente o valor limite desejado entre os 1000 e os 3500 segundos.

Os K_p e k_i foram afinados com valores compreendidos entre '0' e '100' sendo escolhido os valores que apresentavam a menor taxa de rejeição de chamadas. Durante esta afinação não foi observada qualquer situação de instabilidade.

3.5 Simulação do Controlo com dois *Limiares*

Este controlo funciona fixando dois *Limiares* de entrada de novas chamadas, um *Limiar* máximo e um *Limiar* mínimo. A violação do *Limiar* máximo dá origem à rejeição de chamadas. O *Limiar* mínimo pode ser ultrapassado uma vez, sendo que se no segundo seguinte à 1ª violação este limite for novamente violado o algoritmo começa a rejeitar chamadas. A rejeição de chamadas faz-se exactamente do mesmo modo que o algoritmo de controlo com um *Limiar*.

Os *Limiares* poderão ser estáticos, ou ambos dinâmicos. Os *Limiares* dinâmicos são obtidos utilizando controlo *Proporcional* ou *Proporcional-Integrador*.

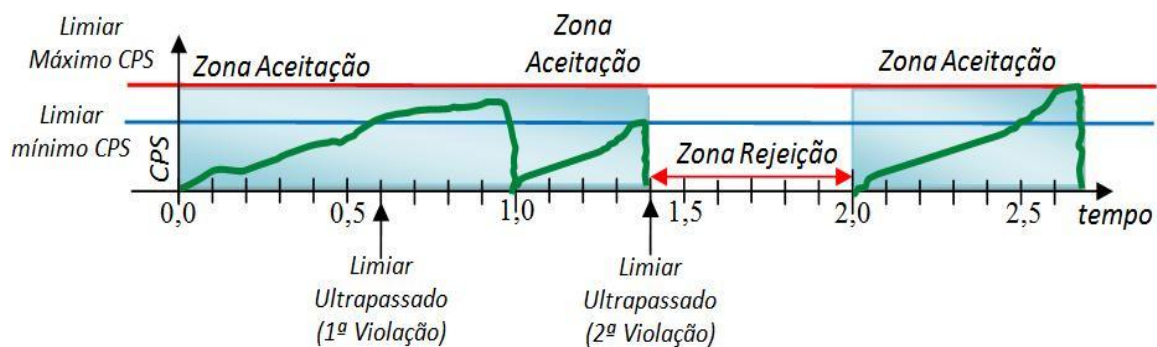


Figura 27 - Esquema de Aceitação e Rejeição do SCF (dois Limiares).

3.5.1 Controlo de um SCF

Os *Limiares* dinâmicos são obtidos usando a mesma malha de controlo da Figura 18, e usando exactamente as mesmas fórmulas descritas em 3.4.1 do controlo *Proporcional* (10), e *Proporcional-Integrador* (14), com a diferença de que elas tem que ser aplicadas ao *Limiar* máximo e também ao *Limiar* mínimo.

	<i>Limiar</i> Máx	<i>Limiar</i> mín	K_p	K_i	Nº Chamadas Chegadas	Nº Chamadas Rejeitadas	% Chamadas Rejeitadas	EAC RAM SCF	EAC CPU SCF
Controlo SCF: Estático	10	9	---	---	31 506	2 799	8,8840 %	0	$1,0422 \times 10^{-4}$
Controlo SCF: Proporcional	21	20	8	---	31 506	2 016	6,3988 %	$5,9304 \times 10^{-5}$	0,0028
Controlo SCF: PI	21	19	4	10	31 506	1 985	6,3004 %	0,0049	0,0115

Tabela 10 - Resultados do Controlo de um SCF.

Os K_p e K_i foram afinados com valores compreendidos entre '0' e '100' sendo escolhido os valores que apresentavam a menor taxa de rejeição de chamadas. Durante esta afinação não foi observada qualquer situação de instabilidade.

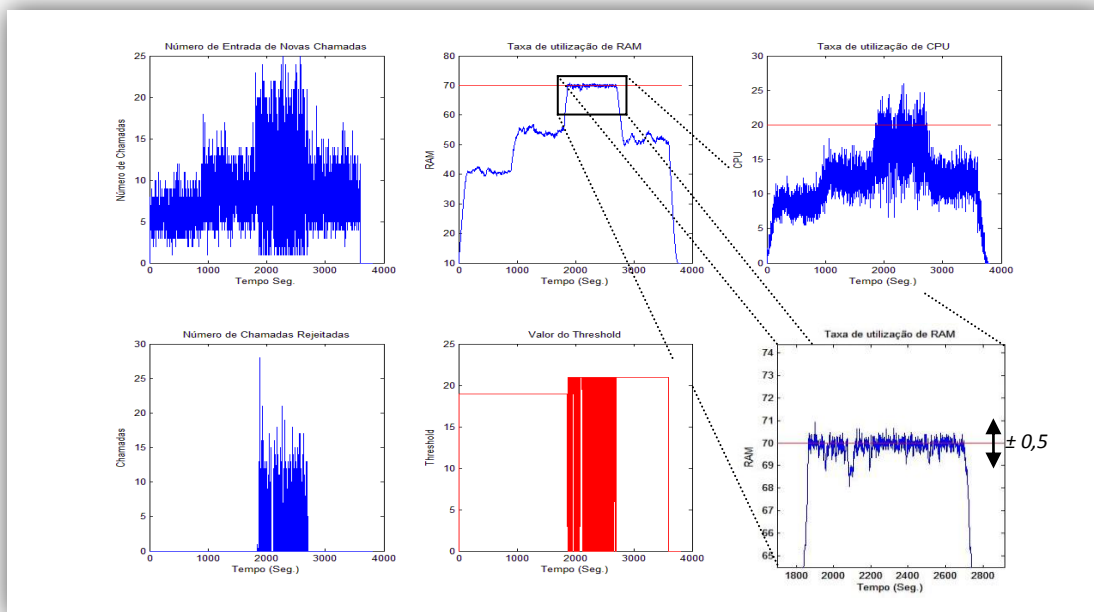


Figura 28 - Simulação do Controlo do SCF com dois Limiares Dinâmicos PI (melhor resultado).

Observando a *Tabela 10* pode-mos dizer que obtemos um controlo com menor rejeição de chamadas utilizando um controlo *Proporcional-Integrador* no *SCF*. Este controlador apresenta também os maiores valores de erro acumulado de *RAM* e *CPU*.

3.5.2 Controlo de um *SCF* e um *SDF*

Os *Limiares* dinâmicos são obtidos usando a mesma malha de controlo da *Figura 21*, e usando exactamente as mesmas fórmulas descritas em 3.4.2 do controlo *Proporcional* (16), e *Proporcional-Integrador* (20), com a diferença de que elas tem que ser aplicadas ao *Limiar* máximo e também ao *Limiar* mínimo.

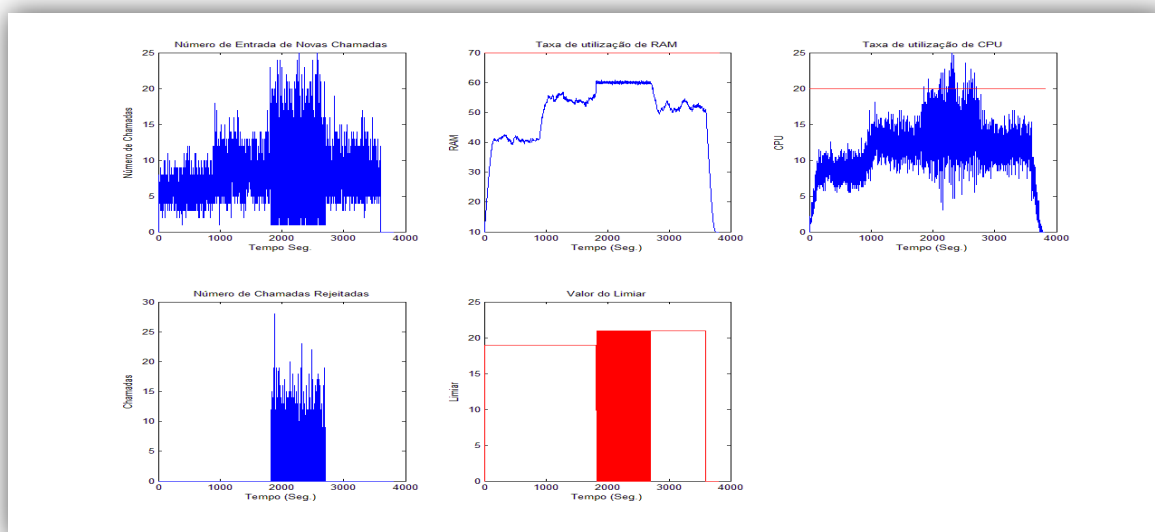


Figura 29 - SCF: Simulação do Controlo do SCF e SDF com dois Limiares Dinâmicos PI (melhor resultado).

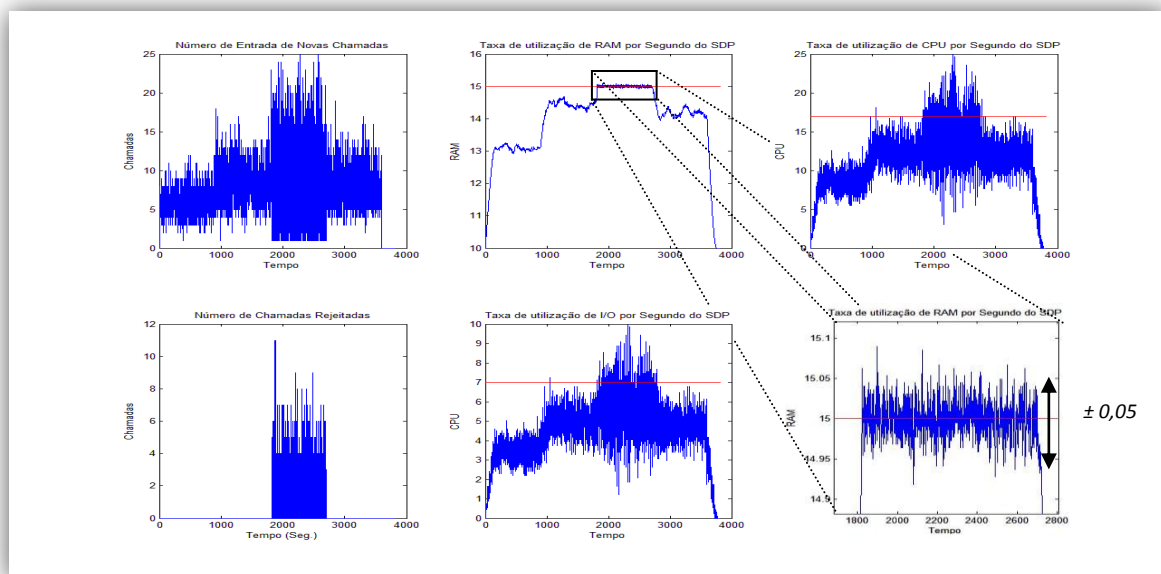


Figura 30 - SDF: Simulação do Controle do SCF e SDF com dois Limiares Dinâmicos PI (melhor resultado).

Em baixo podemos observar uma tabela com os resultados dos diversos controlos do SCF e SDF. Os Limiares, o K_p e K_I do SCF já foram calculados no ponto 3.5.1.

	K_{p_SDF}	K_{I_SDF}	Nº Chamadas Chegadas	Nº Chamadas Rejeitadas	% Chamadas Rejeitadas	EAC RAM SCF	EAC CPU SCF	EAC RAM SDF	EAC CPU SDF	EAC I/O SDF
Controlo SCF: Estático Controlo SDF: Proporcional	35	-----	31 506	4 115	13,0610 %	0	0	$1,6516 \times 10^{-7}$	0,0015	$1,2117 \times 10^{-4}$
Controlo SCF: Estático Controlo SDF: PI	45	3,8	31 506	4 080	12,9499 %	0	0	$4,4483 \times 10^{-6}$	$4,7860 \times 10^{-6}$	$2,2358 \times 10^{-5}$
Controlo SCF: Proporcional Controlo SDF: Proporcional	35	-----	31 506	3 699	11,7406 %	0	$3,9329 \times 10^{-6}$	$9,5293 \times 10^{-7}$	0,0026	$1,9922 \times 10^{-4}$
Controlo SCF: Proporcional Controlo SDF: PI	35	5	31 506	3 665	11,6327 %	0	$2,3204 \times 10^{-4}$	$3,2644 \times 10^{-5}$	0,0117	0,0011
Controlo SCF: PI Controlo SDF: Proporcional	58	-----	31 506	3 685	11,6962 %	0	$7,4725 \times 10^{-5}$	$1,2403 \times 10^{-6}$	0,0034	$2,8931 \times 10^{-4}$
Controlo SCF: PI Controlo SDF: PI	100	75	31 506	3 643	11,5629 %	0	0,0046	$8,3658 \times 10^{-5}$	0,0403	0,0046

Tabela 11 - Resultados do Controle do SCF e SDF (dois Limiares).

Observando a *Tabela 11* podemos constatar que se obtém o menor número de rejeição de chamadas usando controle *PI* no *SCF* e no *SDF*. Esta combinação também apresenta os maiores erros acumulados de *RAM*, *CPU* e *I/O*.

Comparativamente com o controle usando apenas um *Limiar* podemos ver que existe uma diminuição do número das chamadas rejeitadas.

Os K_p e k_I foram afinados com valores compreendidos entre '0' e '100' sendo escolhido os valores que apresentavam a menor taxa de rejeição de chamadas. Durante esta afinação não foi observada qualquer situação de instabilidade.

3.5.3 Controle de quatro *SCF* e um *SDF*

Para obtenção dos *Limiares* dinâmicos é usada a mesma malha de controle da *Figura 24*, e usando exactamente as mesmas fórmulas com o esquema de prioridades (*Tabela 7*) do controle *Proporcional* (21) e *Proporcional-Integrador* (22) descritas no ponto 3.4.3 com a diferença de que elas tem que ser aplicadas ao *Limiar* máximo e também ao *Limiar* mínimo.

Todos os *SCF* terão sempre o mesmo tipo de controle, isto é, ou terão todos controle *Estáticos*, ou *Proporcional*, ou *Proporcional-Integrador*.

	K_{p_SDF}	K_{I_SDF}	Nº Chamadas Chegadas	Nº Chamadas Rejeitadas	% Chamadas Rejeitadas	EAC RAM SDF	EAC CPU SDF	EAC I/O SDF
Controlo SCF: Estático Controlo SDF: Proporcional	85	-----	130 500	35 114	26,9073 %	$5,7403 \times 10^{-5}$	0,0127	0,0020
Controlo SCF: Estático Controlo SDF: PI	20	105	130 500	34 374	26,3402 %	0,0084	0,1141	0,0011
Controlo SCF: Proporcional Controlo SDF: Proporcional	80	-----	130 500	34 080	26,6728 %	$4,2442 \times 10^{-4}$	0,1074	0,0172
Controlo SCF: Proporcional Controlo SDF: PI	40	95	130 500	33 996	26,0506 %	0,0217	0,5241	0,0839
Controlo SCF: PI Controlo SDF: Proporcional	85	-----	130 500	34 754	26,6314 %	$4,9160 \times 10^{-4}$	0,0834	0,0134
Controlo SCF: PI Controlo SDF: PI	80	80	130 500	33 974	26,0337 %	0,0215	0,3705	0,0593

Tabela 12 - SDF: Resultados da Simulação com quatro SCF e um SDF (dois Limiares).

Em baixo podemos observar uma tabela com os resultados dos diversos controlos dos quatro SCF.

	SCF	Limiar Máx	Limiar min	K_p	K_i	Nº Chamadas Chegadas	Nº Chamadas Rejeitadas	% Chamadas Rejeitadas	EAC RAM SCF	EAC CPU SCF
Controlo SCF: Estático Controlo SDF: Proporcional	1º	10	9	---	---	32 505	4 971	15,2930%	0	0
	2º	10	9	---	---	32 750	7 450	22,7481 %	0	0
	3º	10	9	---	---	32 757	10 805	32,9853 %	0	0
	4º	10	9	---	---	32 488	11 888	36,5920 %	0	0
Controlo SCF: Estático Controlo SDF: PI	1º	10	9	---	---	32 505	4 937	15,1884 %	0	0
	2º	10	9	---	---	32 750	7 920	24,1832 %	0	0
	3º	10	9	---	---	32 757	10 325	31,5200 %	0	0
	4º	10	9	---	---	32 488	11 192	34,4496 %	0	0
Controlo SCF: Proporcional Controlo SDF: Proporcional	1º	21	20	8	---	32 505	3 487	10,7276 %	0	0,0087
	2º	21	20	8	---	32 750	6 038	18,4366 %	0	$4,4125 \times 10^{-4}$
	3º	21	20	8	---	32 757	10 586	32,3168 %	0	0
	4º	21	20	8	---	32 488	14 697	45,2382 %	0	0
Controlo SCF: Proporcional Controlo SDF: PI	1º	20	20	8	---	32 505	3 661	11,2629 %	0	0,0059
	2º	20	20	8	---	32 750	7 213	22,0244 %	0	0,0015
	3º	20	20	8	---	32 757	10 544	32,1885 %	0	$1,5689 \times 10^{-5}$
	4º	20	20	8	---	32 488	12 578	38,7158 %	0	0
Controlo SCF: PI Controlo SDF: Proporcional	1º	21	19	2	4,5	32 505	3 544	10,9029 %	0	0,0024
	2º	21	19	2	4,5	32 750	5 966	18,2168 %	0	$2,1572 \times 10^{-4}$
	3º	21	19	2	4,5	32 757	10 483	32,0023 %	0	0
	4º	21	19	2	4,5	32 488	14 761	45,4352 %	0	0
Controlo SCF: PI Controlo SDF: PI	1º	21	19	2	4,5	32 505	3 664	11,2721 %	1,5234	$2,8738 \times 10^{-4}$
	2º	21	19	2	4,5	32 750	7 420	22,6565 %	0	0
	3º	21	19	2	4,5	32 757	10 419	31,8069 %	0	0
	4º	21	19	2	4,5	32 488	12,471	38,3865 %	0	0

Tabela 13 - SCF: Resultados da Simulação com quatro SCF e um SDF (dois Limiares).

Pode-mos dizer que utilizando Controlo *Proporcional* no SDF os dois SCF mais prioritários apresentam menor número de chamadas rejeitadas e os dois menos prioritários apresentam maior número de chamadas rejeitadas do que usando controlo *Proporcional-Integrador*.

Neste caso, e relativamente ao controlo usando um *Limiar*, existe aumento de chamadas rejeitadas nas combinações que usam *SDF* com controlo *Proporcional* (excepção para o caso em que os *SCF* utilizam controlo Estático). A combinação *PI* no *SCF* e *SDF* apresenta nesta simulação o valor mais baixo de rejeição de chamadas e também valores mais baixos de rejeição em comparação com o controlo com um *Limiar*.

Os K_p e k_i foram afinados com valores compreendidos entre '0' e '100' sendo escolhido os valores que apresentavam a menor taxa de rejeição de chamadas. Durante esta afinação não foi observada qualquer situação de instabilidade.

Pode-se constatar na *Figura 31* que existe muita mais rejeição de chamadas nos *SCF* menos prioritários.

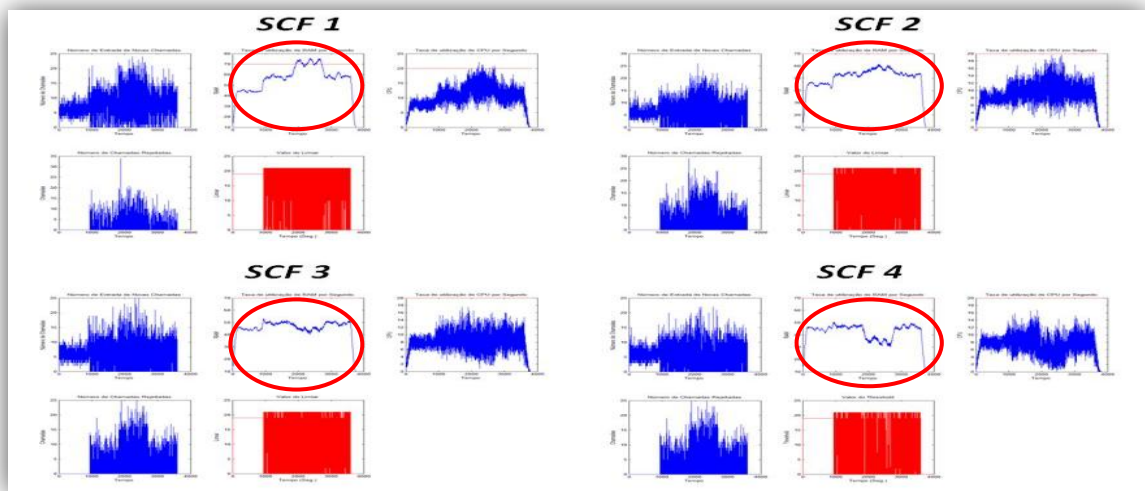


Figura 31 - SDF: Simulação com Controlo PI nos quatro SCF e no SDF (dois Limiares: melhor resultado).

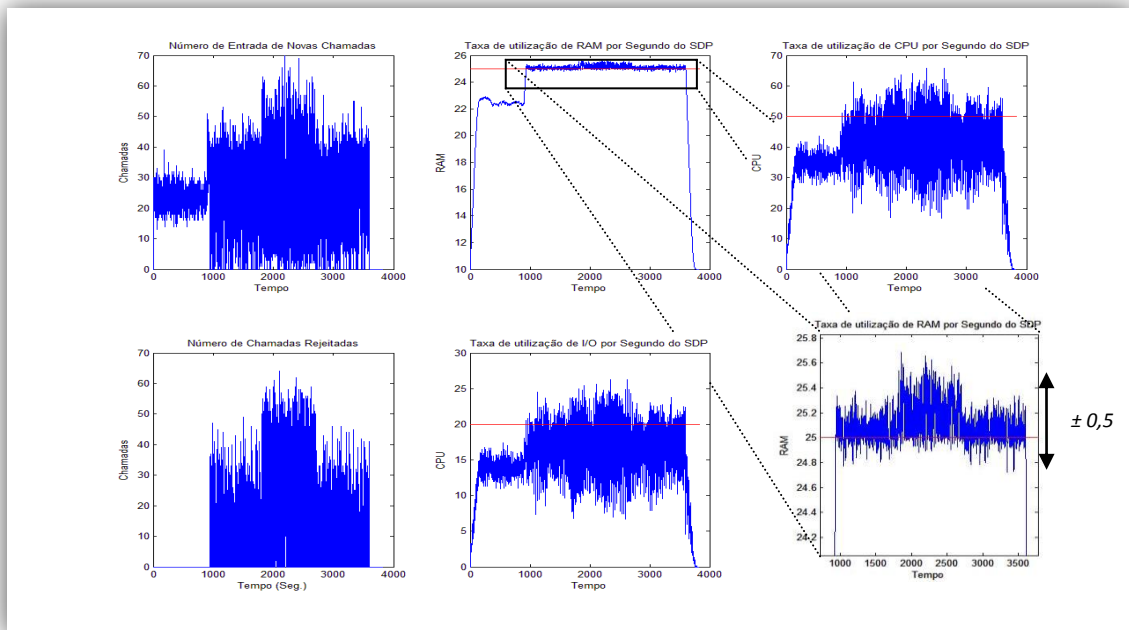


Figura 32 - SDF: Simulação com Controlo PI nos quatro SCF e no SDF (dois Limiares: melhor resultado)

Como podemos ver pela *Figura 32*, podemos dizer que o sistema se encontra controlado, pelo menos na Taxa de Ocupação de RAM.

3.6 Controlo com um *Limiar Estático* e um *Limiar Dinâmico*

O Funcionamento deste controlo é semelhante ao controlo usando dois *Limiares* (*Figura 27*). A única diferença é que o *Limiar* máximo é fixo, e o *Limiar* mínimo é dinâmico.

3.6.1 Controlo de um SCF

O *Limiar* mínimo é obtido usando a mesma malha de controlo da *Figura 18*, e usando as fórmulas do controlo *Proporcional* (10) e *Proporcional-Integrador* (14) descritas no ponto 3.4.1.

	<i>Limiar Máx</i>	<i>Limiar mín</i>	K_p	K_i	Nº Chamadas Chegadas	Nº Chamadas Rejeitadas	% Chamadas Rejeitadas	EAC RAM SCF	AEC CPU SCF
Controlo SCF: Proporcional	23	22	9	-----	31 506	1 994	6,3290 %	0,0014	0,198
Controlo SCF: PI	21	19	4	10	31 506	1 967	6,2433 %	0,0107	0,0233

Tabela 14 - Resultados do Controlo de um SCF (um Limiar Estático e um Limiar Dinâmico).

Podemos ver na *Tabela 14* que o controlo usando um controlador *PI* apresenta a menor percentagem de chamadas rejeitadas, assim como os maiores erros acumulados de RAM e CPU como seria de esperar e cuja causa também já foi explicada anteriormente.

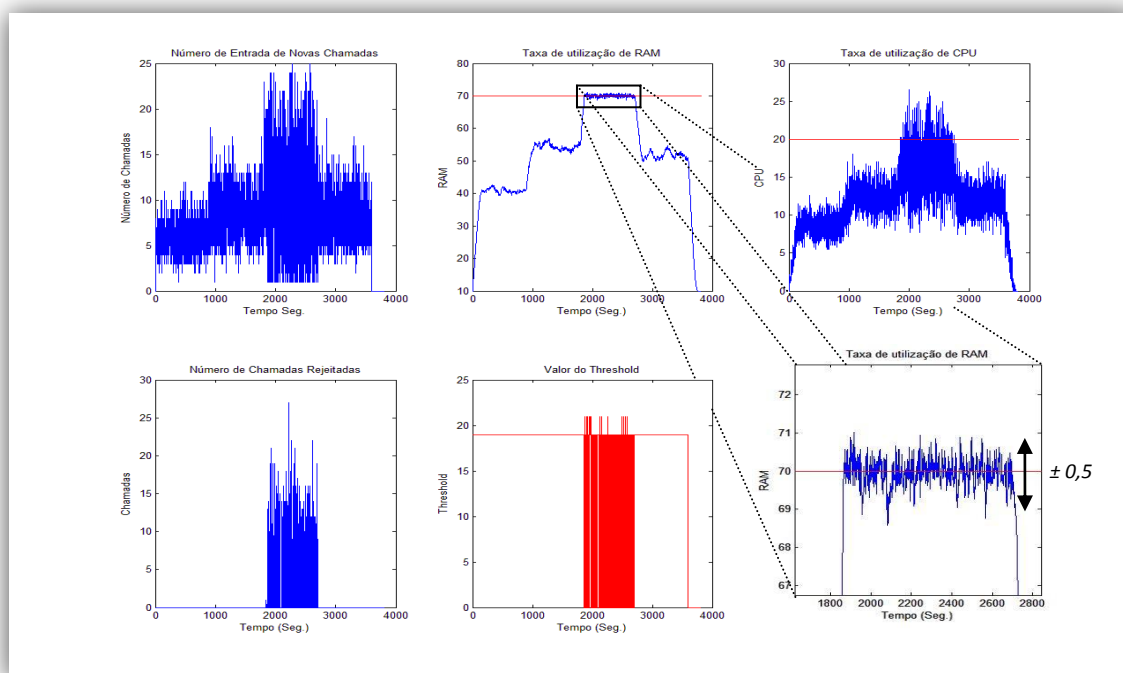


Figura 33 - Simulação do Controlo do SCF com um Limiar Estático e um Limiar Dinâmico PI (melhor resultado).

Os K_p e k_i foram afinados com valores compreendidos entre '0' e '100' sendo escolhido os valores que apresentavam a menor taxa de rejeição de chamadas. Durante esta afinação não foi observada qualquer situação de instabilidade.

3.6.2 Controlo de um SCF e um SDF

O *Limiar* dinâmico é obtidos usando a malha de controlo da *Figura 21*, e usando as fórmulas do controlo *Proporcional* (16) e *Proporcional-Integrador* (20) descritas no ponto 3.4.2.

Em baixo podemos observar uma tabela com os resultados dos diversos controlos do SCF e SDF. Os *Limiares*, o K_p e K_i do SCF já foram calculados no ponto 3.5.1 e 3.6.1.

	K_{p_SDF}	K_{i_SDF}	Nº Chamadas Chegadas	Nº Chamadas Rejeitadas	% Chamadas Rejeitadas	EAC RAM SCF	EAC CPU SCF	EAC RAM SDF	EAC CPU SDF	EAC I/O SDF
Controlo SCF: Estático Controlo SDF: Proporcional	60	-----	31 506	4 086	12,9690%	0	0	$5,8114 \times 10^{-6}$	0,0013	$8,5216 \times 10^{-5}$
Controlo SCF: Estático Controlo SDF: PI	95	75	31 506	4 062	12,8928 %	0	0	$5,3953 \times 10^{-5}$	0,0021	$1,5293 \times 10^{-4}$
Controlo SCF: Proporcional Controlo SDF: Proporcional	40	-----	31 506	3 678	11,6740 %	0	0,0019	$7,8369 \times 10^{-5}$	0,0270	0,0029
Controlo SCF: Proporcional Controlo SDF: PI	45	65	31 506	3 628	11,5153 %	0	0,0044	$3,5760 \times 10^{-4}$	0,0452	0,0051
Controlo SCF: PI Controlo SDF: Proporcional	50	-----	31 506	3 685	11,6962 %	0	0,0023	$8,9974 \times 10^{-5}$	0,0339	0,0037
Controlo SCF: PI Controlo SDF: PI	40	85	31 506	3 624	11,5026 %	0	0,0051	$3,2780 \times 10^{-4}$	0,0488	0,0056

Tabela 15 - Resultados do Controlo do SCF e SDF (um Limiar Estático e um Limiar Dinâmico).

Observando a *Tabela 15* podemos constatar que se obtém o menor número de rejeição de chamadas usando controlo *PI* no SCF e no SDF. Esta combinação também apresenta os maiores erros acumulados de RAM, CPU e I/O.

Os K_p e k_i foram afinados com valores compreendidos entre '0' e '100' sendo escolhido os valores que apresentavam a menor taxa de rejeição de chamadas. Durante esta afinação não foi observada qualquer situação de instabilidade.

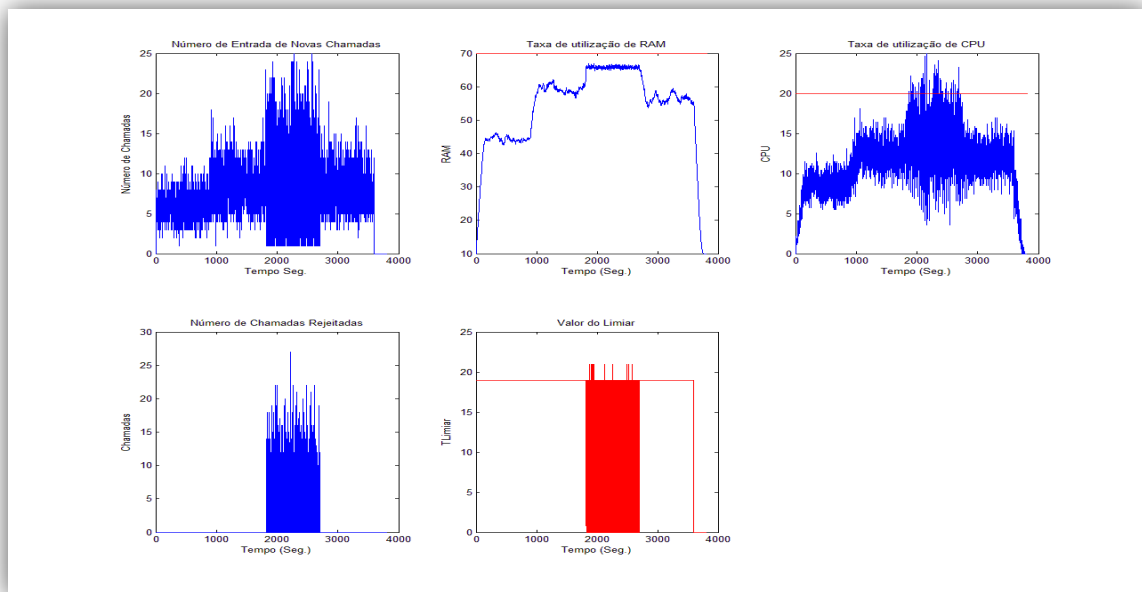


Figura 34 - SCF: Simulação do Controle do SCF e SDF com um Limiar Estático e um Limiar Dinâmico PI (melhor resultado).

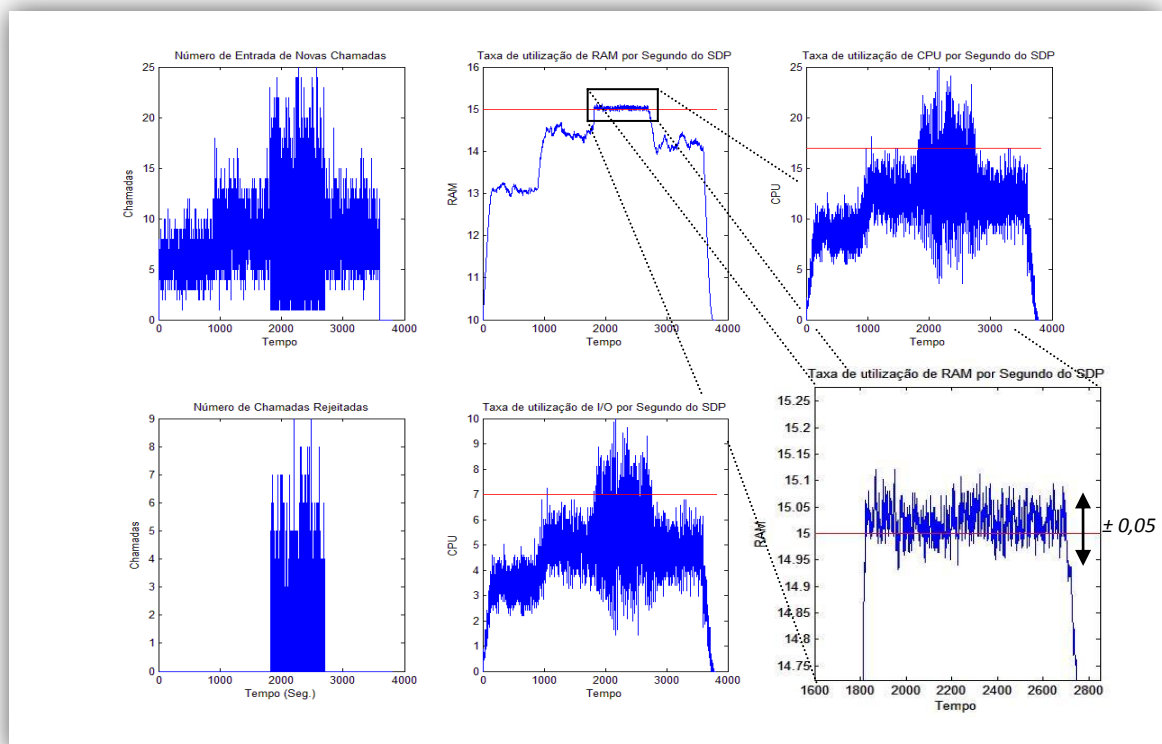


Figura 35 - SDF: Simulação do Controle do SCF e SDF com um Limiar Estático e 1 Limiar Dinâmico PI (melhor resultado).

3.6.3 Controlo de quatro SCF e um SDF

Para obtenção do *Limiar* dinâmico é usada a mesma malha de controlo da *Figura 24*, e usando as fórmulas com esquema de prioridades (*Tabela 7*) do controlo *Proporcional* (21) e *Proporcional-Integrador* (22) descritas no ponto 3.4.3.

Todos os *SCF* terão sempre o mesmo tipo de controlo, isto é, ou terão todos controlo *Estáticos*, ou *Proporcional*, ou *Proporcional-Integrador*.

Na *Figura 36* pode-se constatar que existe muita mais rejeição de chamadas nos *SCF* menos prioritários.

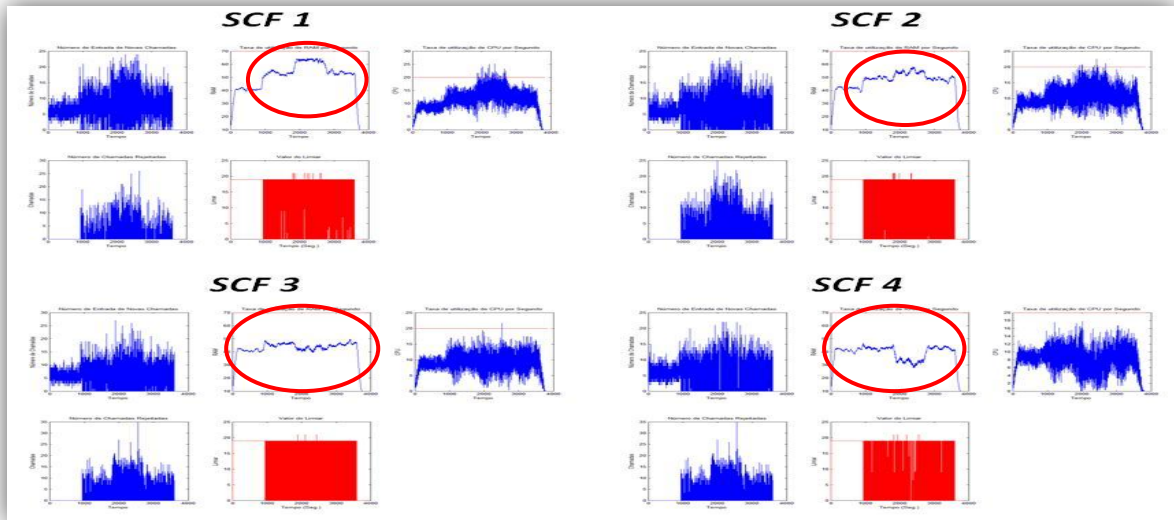


Figura 36 - SCF: Simulação com Controlo PI nos quatro SCF e no SDF (um Limiar Estático e um Limiar Dinâmico).

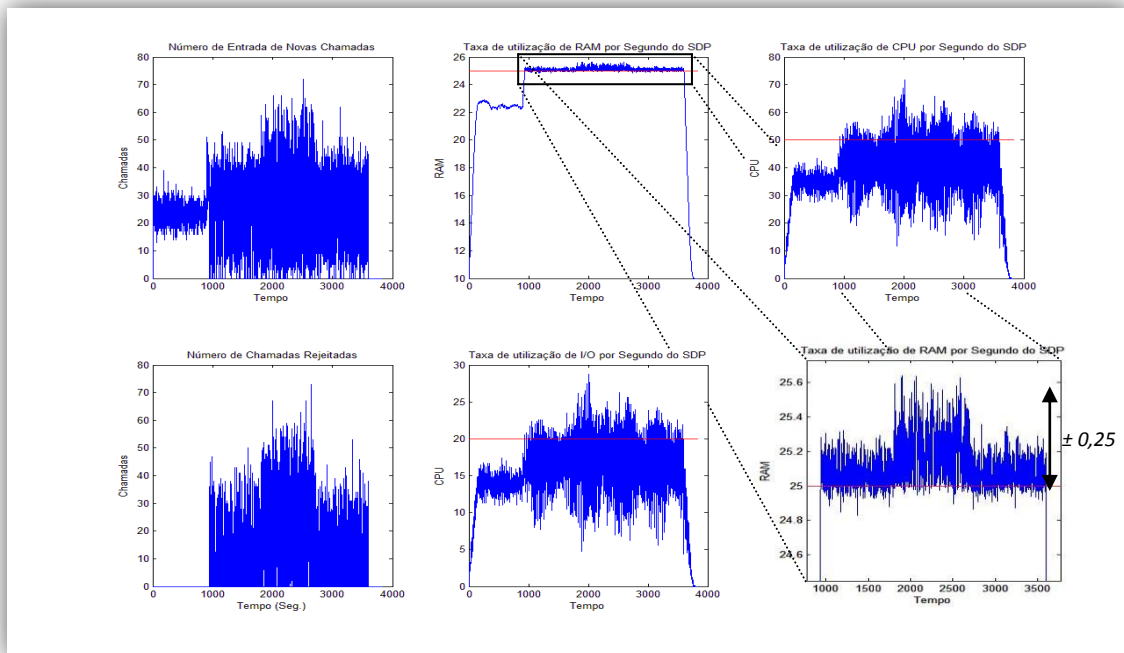


Figura 37 - SDF: Simulação com Controlo PI nos quatro SCF e no SDF (um Limiar Estático e um Limiar Dinâmico).

Como podemos ver pela *Figura 37*, podemos dizer que o sistema se encontra controlado, pelo menos na Taxa de Ocupação de RAM.

Em baixo podemos observar uma tabela com os resultados dos diversos controlos dos quatro SCF.

	SCF	Limiar Máx	Limiar mín	K_p	K_i	Nº Chamadas Chegadas	Nº Chamadas Rejeitadas	% Chamadas Rejeitadas	EAC RAM SCF	EAC CPU SCF
Controlo SCF: Estático Controlo SDF: Proporcional	1º	10	9	----	---	32 505	4 988	15,3453 %	0	0
	2º	10	9	----	---	32 750	7 135	21,7863 %	0	0
	3º	10	9	----	---	32 757	10 070	30,7415 %	0	0
	4º	10	9	----	---	32 488	12 532	38,5742 %	0	0
Controlo SCF: Estático Controlo SDF: PI	1º	10	9	----	---	32 505	5 286	16,2621 %	0	0
	2º	10	9	----	---	32 750	7 851	23,9725 %	0	0
	3º	10	9	----	---	32 757	10 094	30,8148 %	0	0
	4º	10	9	----	---	32 488	10 973	33,7755 %	0	0
Controlo SCF: Proporcional Controlo SDF: Proporcional	1º	23	22	8	---	32 505	3 078	9,4693 %	$4,7028 \times 10^{-7}$	0,0096
	2º	23	22	8	---	32 750	5 958	18,1924 %	0	$1,0198 \times 10^{-4}$
	3º	23	22	8	---	32 757	10 283	31,3918 %	0	0
	4º	23	22	8	---	32 488	15 348	47,2421 %	0	0
Controlo SCF: Proporcional Controlo SDF: PI	1º	23	22	8	---	32 505	3 686	11,3398 %	0	0,0034
	2º	23	22	8	---	32 750	7 088	21,6427 %	0	$3,5888 \times 10^{-4}$
	3º	23	22	8	---	32 757	10 326	31,5230 %	0	$2,8632 \times 10^{-4}$
	4º	23	22	8	---	32 488	12 845	39,5377 %	0	0
Controlo SCF: PI Controlo SDF: Proporcional	1º	21	19	2	4,5	32 505	3 416	10,5092 %	0	0,0036
	2º	21	19	2	4,5	32 750	6 053	18,4824 %	0	$8,6680 \times 10^{-4}$
	3º	21	19	2	4,5	32 757	10 189	31,1048 %	0	0
	4º	21	19	2	4,5	32 488	14 901	45,8662 %	0	0
Controlo SCF: PI Controlo SDF: PI	1º	21	19	2	4,5	32 505	4 076	12,5396 %	0	0,0056
	2º	21	19	2	4,5	32 750	7 166	21,8809 %	0	$2,4710 \times 10^{-4}$
	3º	21	19	2	4,5	32 757	10 145	30,9705 %	0	$7,0599 \times 10^{-5}$
	4º	21	19	2	4,5	32 488	12 519	38,5342 %	0	0

Tabela 16 - SCF: Resultados da Simulação com quatro SCF e um SDF (um Limiar Estático e um Limiar Dinâmico).

	K_{p_SDF}	K_{I_SDF}	Nº Chamadas Chegadas	Nº Chamadas Rejeitadas	% Chamadas Rejeitadas	EAC RAM SDF	EAC CPU SDF	EAC I/O SDF
Controlo SCF: Estático Controlo SDF: Proporcional	85	-----	130 500	34 725	26,6092 %	$1,5162 \times 10^{-4}$	0,0163	0,0026
Controlo SCF: Estático Controlo SDF: PI	40	70	130 500	34 204	26,2100 %	0,0095	0,1563	0,0250
Controlo SCF: Proporcional Controlo SDF: Proporcional	30	-----	130 500	34 667	26,5648 %	$4,9030 \times 10^{-4}$	0,0887	0,0142
Controlo SCF: Proporcional Controlo SDF: PI	50	65	130 500	42 326	26,0115 %	0,0220	0,3512	0,0562
Controlo SCF: PI Controlo SDF: Proporcional	80	-----	130 500	34 559	26,4820 %	$6,2834 \times 10^{-4}$	0,0463	0,0074
Controlo SCF: PI Controlo SDF: PI	50	65	130 500	33 906	25,9816 %	0,0226	0,3812	0,0610

Tabela 17- SDF: Resultados da Simulação com quatro SCF e um SDF (um Limiar Estático e um Limiar Dinâmico)

Podemos dizer que utilizando Controlo *Proporcional* no SDF os dois SCF mais prioritários apresentam menor número de chamadas rejeitadas e os dois menos prioritários apresentam maior número de chamadas rejeitadas do que usando controlo *Proporcional-Integrador*. Repare-se ainda que usando o controlo com dois *Limiares* há um aumento da rejeição de chamadas nos SCF mais prioritários em relação ao controlo com um *Limiar* (excepção para o caso em que os SCF utilizam controlo Estático).

A combinação *PI* no SCF e SDF apresenta o valor mais baixo de rejeição de chamadas nesta simulação e também valores mais baixos de rejeição em comparação aos controlos estudados anteriormente.

Os K_p e k_I foram afinados com valores compreendidos entre '0' e '100' sendo escolhido os valores que apresentavam a menor taxa de rejeição de chamadas. Durante esta afinação não foi observada qualquer situação de instabilidade.

3.7 Simulação do Controlo com um *Limiar* de Eventos

O Funcionamento deste controlo é semelhante ao controlo usando um *Limiar*, com a diferença de que este *Limiar* é um limite máximo para eventos e não para chamadas. Quando este limiar é ultrapassado, dá lugar à rejeição de chamadas tal e qual como os algoritmos anteriores. O objectivo deste controlo é dar mais importância ao número de eventos que cada máquina pode processar em vez do número de chamadas activas que cada máquina pode ter, dando deste modo um peso maior ao *CPU*.

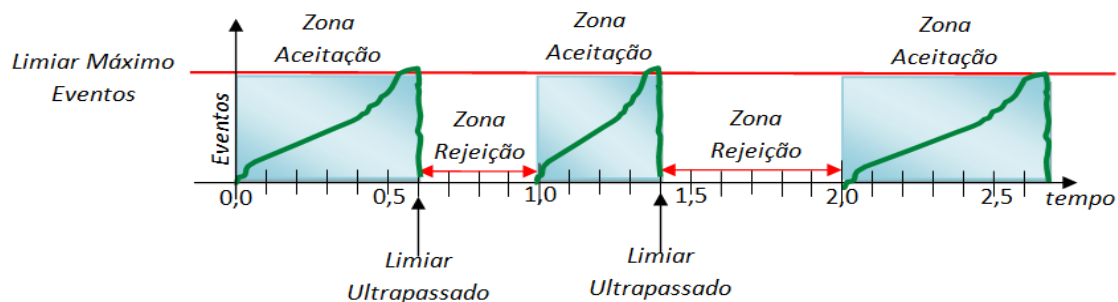


Figura 38- Esquema de Aceitação e Rejeição do SCF (Limiar eventos).

3.7.1 Controlo de um SCF

O *Limiar* dinâmico é obtido usando a mesma malha de controlo da Figura 18 usando as mesmas fórmulas do controlo *Proporcional* (10) e *Proporcional-Integrador* (14) descritas no ponto 3.4.1.

	<i>Limiar</i>	K_p	K_i	Nº Chamadas Chegadas	Nº Chamadas Rejeitadas	% Chamadas Rejeitadas	EAC RAM SCF	EAC CPU SCF
Controlo SCF: Estático	41	-----	-----	31 506	4 336	13,7625 %	$3,2485 \times 10^{-5}$	0,0117
Controlo SCF: Proporcional	62	10	-----	31 506	2 041	6,4781%	$5,1447 \times 10^{-4}$	0,0011
Controlo SCF: PI	63	2	3	31 506	2 028	6,4369 %	0,0026	0,0043

Tabela 18 - Resultados do Controlo de um SCF (um Limiar Eventos).

Podemos ver na Tabela 18 que o controlo usando um controlador *PI* apresenta a menor percentagem de chamadas rejeitadas, assim como os maiores erros acumulados de *RAM* e *CPU*. Verificamos que este controlo apresenta nesta simulação maior taxa de rejeição de chamadas do que os outros algoritmos de controlo estudados nos pontos anteriores.

Os K_p e k_i foram afinados com valores compreendidos entre '0' e '100' sendo escolhido os valores que apresentavam a menor taxa de rejeição de chamadas. Durante esta afinação não foi observada qualquer situação de instabilidade.

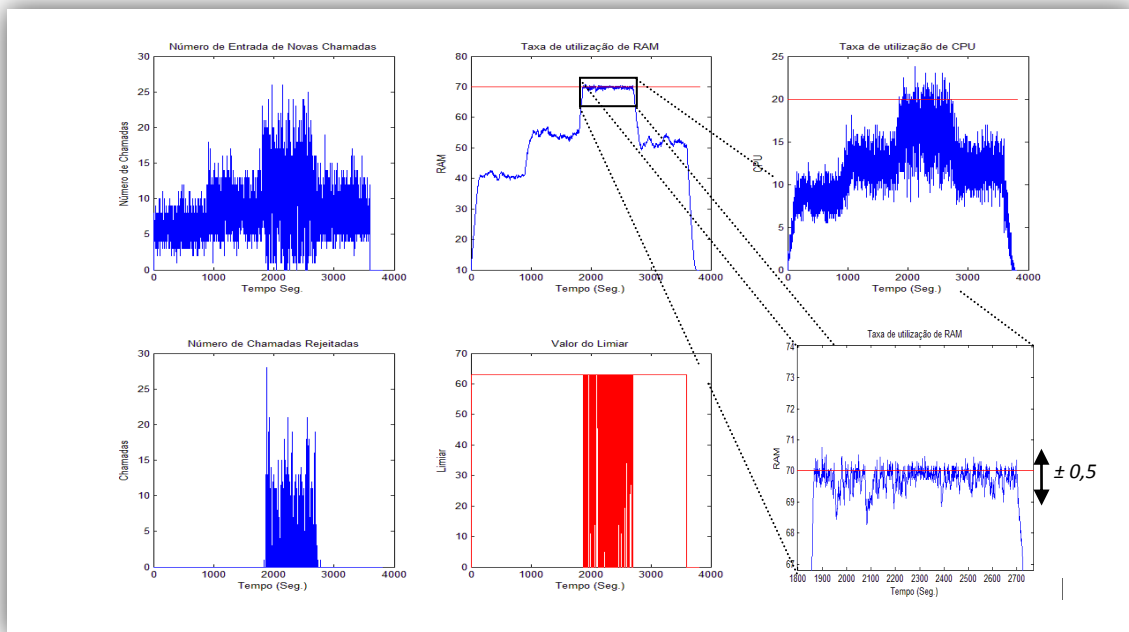


Figura 39 - Simulação do Controle do SCF com um Limiar Eventos PI (melhor resultado).

3.7.2 Controle de um SCF e um SDF

O *Limiar* dinâmico é obtido usando a malha de controle da Figura 21, e as fórmulas do controle *Proporcional* (16) e *Proporcional-Integrador* (20) descritas no ponto 3.4.2.

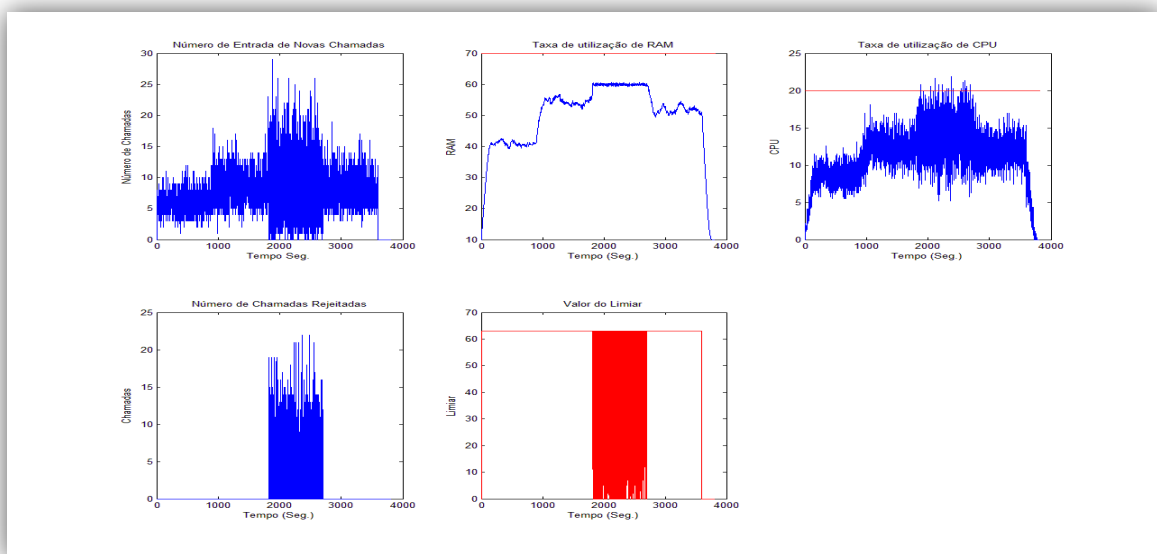


Figura 40 - SCF: Simulação do Controle do SCF e SDF com um Limiar Eventos Dinâmico PI (melhor resultado).

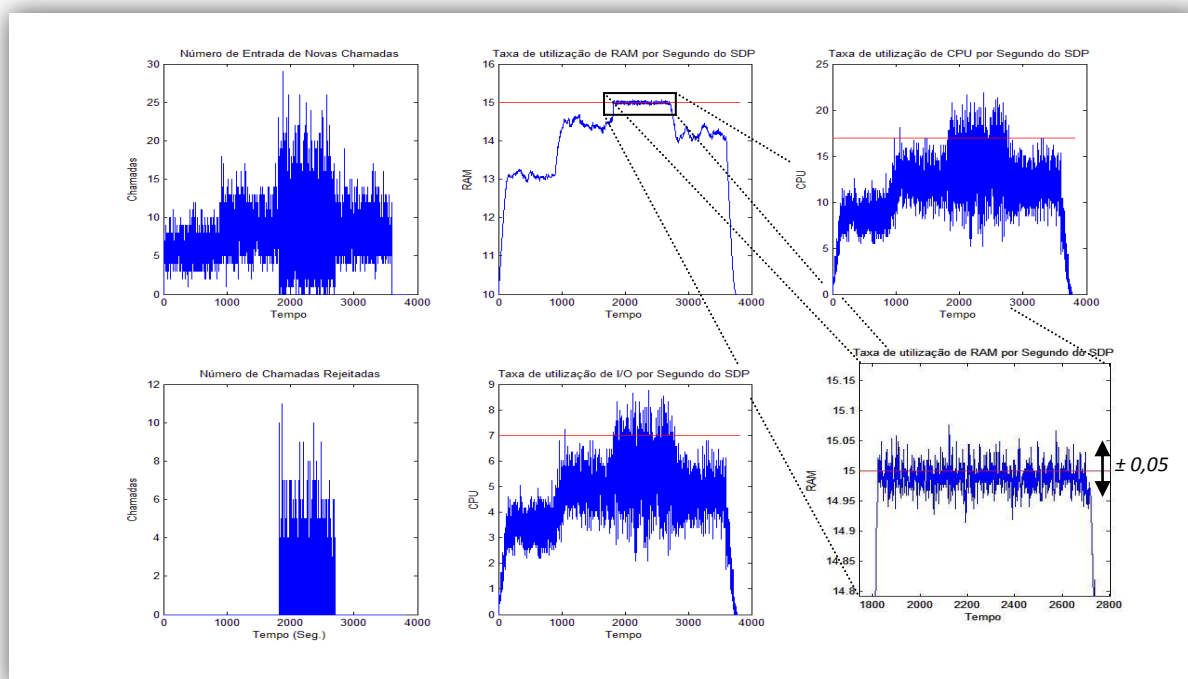


Figura 41 - SDF: Simulação do Controle do SCF e SDF com um Limiar Eventos Dinâmico PI (melhor resultado).

	K_{p_SDF}	K_{I_SDF}	Nº Chamadas Chegadas	Nº Chamadas Rejeitadas	% Chamadas Rejeitadas	EAC RAM SCF	EAC CPU SCF	EAC RAM SCF	EAC CPU SDF	EAC I/O SDF
Controlo SCF: Estático Controlo SDF: Proporcional	60	-----	31 506	5 505	14,2989%	0	$1,9664 \times 10^{-5}$	$4,6488 \times 10^{-8}$	0,0017	$1,6734 \times 10^{-4}$
Controlo SCF: Estático Controlo SDF: PI	95	75	31 506	4 462	14,1624 %	0	0,0551	$1,1220 \times 10^{-6}$	0,4744	0,0548
Controlo SCF: Proporcional Controlo SDF: Proporcional	40	-----	31 506	3 699	11,7406 %	0	$6,2926 \times 10^{-5}$	$2,5133 \times 10^{-7}$	0,0031	$2,7234 \times 10^{-4}$
Controlo SCF: Proporcional Controlo SDF: PI	45	65	31 506	3 681	11,6835 %	0	$1,1602 \times 10^{-4}$	$2,3022 \times 10^{-5}$	0,0085	$7,5006 \times 10^{-4}$
Controlo SCF: PI Controlo SDF: Proporcional	50	-----	31 506	3 707	11,7670 %	0	$1,4748 \times 10^{-4}$	$5,3469 \times 10^{-7}$	0,0041	$3,7040 \times 10^{-4}$
Controlo SCF: PI Controlo SDF: PI	40	85	31 506	3 676	11,6676 %	0	$4,1689 \times 10^{-4}$	$4,2210 \times 10^{-5}$	0,0181	0,0018

Tabela 19 - Resultados do Controle do SCF e SDF (Limiar Eventos).

Observando a *Tabela 19* verificamos que se obtém o menor número de rejeição de chamadas usando controlo *PI* no *SCF* e no *SDF*, apresentando esta combinação também os maiores erros acumulados de *RAM*, *CPU* e *I/O*. Em comparação com os outros algoritmos de controlo estudados nos pontos anteriores podemos dizer que há um aumento do número de chamadas rejeitadas.

Os K_p e k_i foram afinados com valores compreendidos entre '0' e '100' sendo escolhido os valores que apresentavam a menor taxa de rejeição de chamadas. Durante esta afinação não foi observada qualquer situação de instabilidade.

3.7.3 Controlo de quatro *SCF* e um *SDF*

Para obtenção do *Limiar* dinâmico é usada a mesma malha de controlo da *Figura 24*, e usando as mesmas fórmulas com o esquema de prioridades (*Tabela 7*) do controlo *Proporcional* (21) e *Proporcional-Integrador* (22) descritas no ponto 3.4.3.

Todos os *SCF* terão sempre o mesmo tipo de controlo, isto é, ou terão todos controlo *Estáticos*, ou *Proporcional*, ou *Proporcional-Integrador*.

	K_{p_SDF}	K_{i_SDF}	Nº Chamadas Chegadas	Nº Chamadas Rejeitadas	% Chamadas Rejeitadas	EAC RAM SDF	EAC CPU SDF	EAC I/O SDF
Controlo SCF: Estático Controlo SDF: Proporcional	80	-----	130 500	35 321	27,0659 %	$8,6900 \times 10^{-5}$	0,0163	0,0026
Controlo SCF: Estático Controlo SDF: PI	40	105	130 500	34 911	26,7517 %	$6,8064 \times 10^{-4}$	0,0440	0,0070
Controlo SCF: Proporcional Controlo SDF: Proporcional	70	-----	130 500	35 011	26,8284 %	$3,5797 \times 10^{-4}$	0,1070	0,0171
Controlo SCF: Proporcional Controlo SDF: PI	96	85	130 500	34 058	26,0981 %	0,0215	0,2470	0,0395
Controlo SCF: PI Controlo SDF: Proporcional	80	-----	130 500	35 022	26,8368 %	$4,4317 \times 10^{-4}$	0,0866	0,0139
Controlo SCF: PI Controlo SDF: PI	35	25	130 500	34 047	26,0897 %	0,0211	0,3875	0,0620

Tabela 20 - SDF: Resultados da Simulação com quatro SCF e um SDF (Limiar Eventos).

Em baixo podemos observar uma tabela com os resultados dos diversos controlos dos 4 SCF.

	SCF	Limiar	K_p	K_i	Nº Chamadas Chegadas	Nº Chamadas Rejeitadas	% Chamadas Rejeitadas	EAC RAM SCF	EAC CPU SCF
Controlo SCF: Estático Controlo SDF: Proporcional	1º	41	----	---	32 505	5 694	17,5173 %	0	0
	2º	41	----	---	32 750	7 696	23,4992 %	0	0
	3º	41	----	---	32 757	10 578	32,2923 %	0	0
	4º	41	----	---	32 488	11 353	34,9452 %	0	0
Controlo SCF: Estático Controlo SDF: PI	1º	41	----	---	32 505	5 780	17,7819 %	0	$1,9611 \times 10^{-6}$
	2º	41	----	---	32 750	7 817	23,8687 %	0	0
	3º	41	----	---	32 757	10 532	31,6024 %	0	0
	4º	41	----	---	32 488	10 962	33,7417 %	0	0
Controlo SCF: Proporcional Controlo SDF: Proporcional	1º	21	8	---	32 505	3 704	11,3952 %	0	0,0045
	2º	21	8	---	32 750	6 862	20,9527 %	0	$2,5690 \times 10^{-5}$
	3º	21	8	---	32 757	10 914	33,3181 %	0	0
	4º	21	8	---	32 488	13 531	41,6492 %	0	0
Controlo SCF: Proporcional Controlo SDF: PI	1º	20	8	---	32 505	3 782	11,6351 %	0	$8,5896 \times 10^{-4}$
	2º	20	8	---	32 750	7 383	22,5435 %	0	$1,78468 \times 10^{-4}$
	3º	20	8	---	32 757	10 622	32,4267 %	0	0
	4º	20	8	---	32 488	12 271	37,7709 %	0	0
Controlo SCF: PI Controlo SDF: Proporcional	1º	21	2	4,5	32 505	3 677	11,3121 %	0	$7,2561 \times 10^{-4}$
	2º	21	2	4,5	32 750	6 683	20,4061 %	0	$3,3142 \times 10^{-4}$
	3º	21	2	4,5	32 757	11 076	33,8126 %	0	0
	4º	21	2	4,5	32 488	13 586	41,8185 %	0	0
Controlo SCF: PI Controlo SDF: PI	1º	21	2	4,5	32 505	3 687	11,3429 %	0	0,0056
	2º	21	2	4,5	32 750	7 295	22,2748 %	0	$3,4515 \times 10^{-4}$
	3º	21	2	4,5	32 757	10 801	32,9731 %	0	0
	4º	21	2	4,5	32 488	12 264	37,7493 %	0	0

Tabela 21 - SCF: Resultados da Simulação com quatro SCF e um SDF (Limiar Eventos).

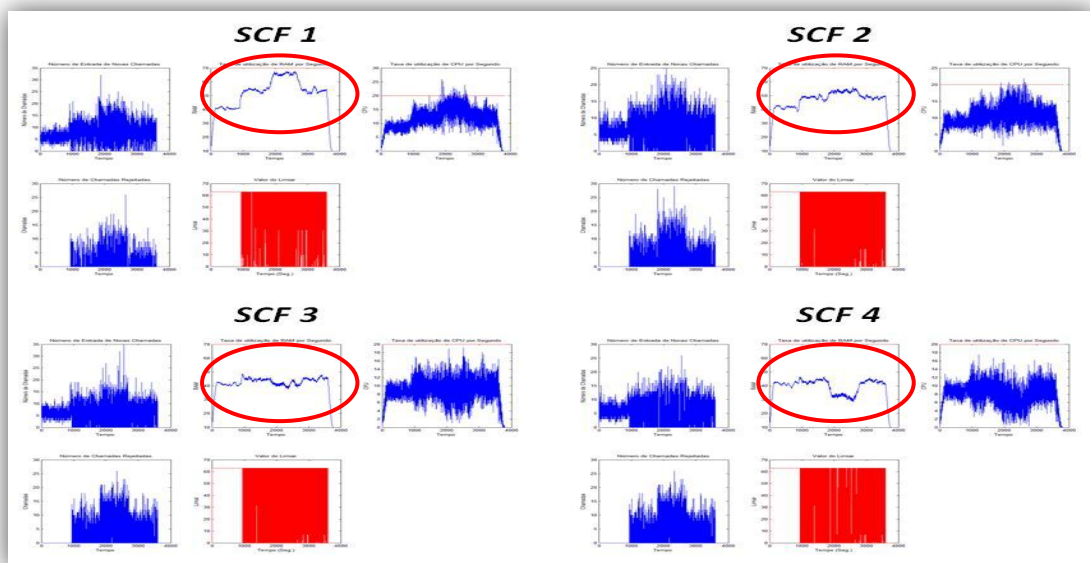


Figura 42 – SCF: Controle PI nos quatro SCF e no SDF com um Limiar Eventos Dinâmico (melhor resultado).

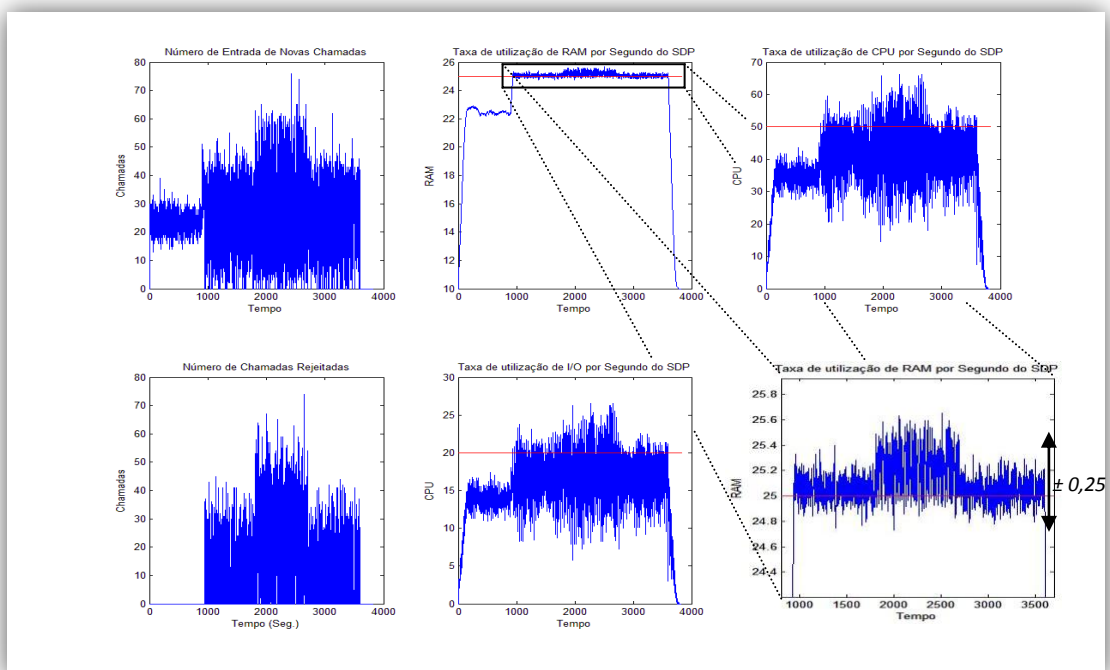


Figura 43 - SDF: Controle PI nos quatro SCF e no SDF com um Limiar Eventos Dinâmico (melhor resultado).

Utilizando Controle PI nos SCF e no SDF obtemos o valor mais baixo de rejeição de chamadas desta simulação. De notar que com este controlo foram obtidos resultados piores que os outros algoritmos estudados acima.

Os K_p e k_i foram afinados com valores compreendidos entre '0' e '100' sendo escolhido os valores que apresentavam a menor taxa de rejeição de chamadas. Durante esta afinação não foi observada qualquer situação de instabilidade.

3.8 Simulação do Controle com Janela Temporal

Este controle utiliza janelas temporais de 1 Segundo para controle de carga. O algoritmo de controle vai atribuir, consoante a carga do sistema, uma parte ou a totalidade dessa janela para aceitação de chamadas. A Largura dessa janela é calculado de duas maneiras, ou usando controle *Proporcional* ou *Proporcional-Integrador*.

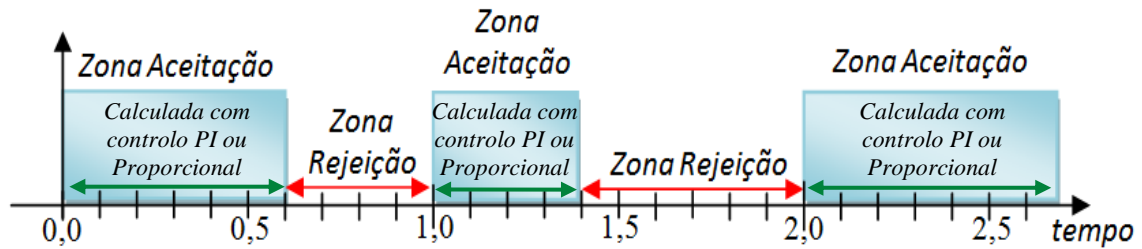


Figura 44 – Esquema de Controle com Janelas Temporais.

3.8.1 Controle de um SCF

Para a obtenção da largura da janela é usada a mesma malha de controle da Figura 18, sendo a largura total da janela igual a 1 Segundo.

Controle Proporcional:

$$Janela(n) = eP_{SCF}(n) \times \text{Largura Total Janela (1 Seg.)} \quad (24)$$

Controle Proporcional-Integrador:

$$Janela(n) = ePI_{SCF}(n) \times \text{Largura Total Janela (1 Seg.)} \quad (25)$$

	K_p	K_i	Nº Chamadas Chegadas	Nº Chamadas Rejeitadas	% Chamadas Rejeitadas	EAC RAM SCF	EAC CPU SCF
Controle SCF: Proporcional	5	-----	31 506	2 728	8,6587 %	$3,8316 \times 10^{-6}$	0,0298
Controle SCF: PI	1	3	31 506	2 006	6,3670 %	0,0052	0,0208

Tabela 22 - Resultados do Controle de um SCF (Janelas Temporais).

Existe grande disparidade de valores entre o controle *Proporcional* e o *PI* apresentando este último, os valores mais baixos de rejeição de chamadas.

Os K_p e k_i foram afinados com valores compreendidos entre '0' e '100' sendo escolhido os valores que apresentavam a menor taxa de rejeição de chamadas. Durante esta afinação não foi observada qualquer situação de instabilidade.

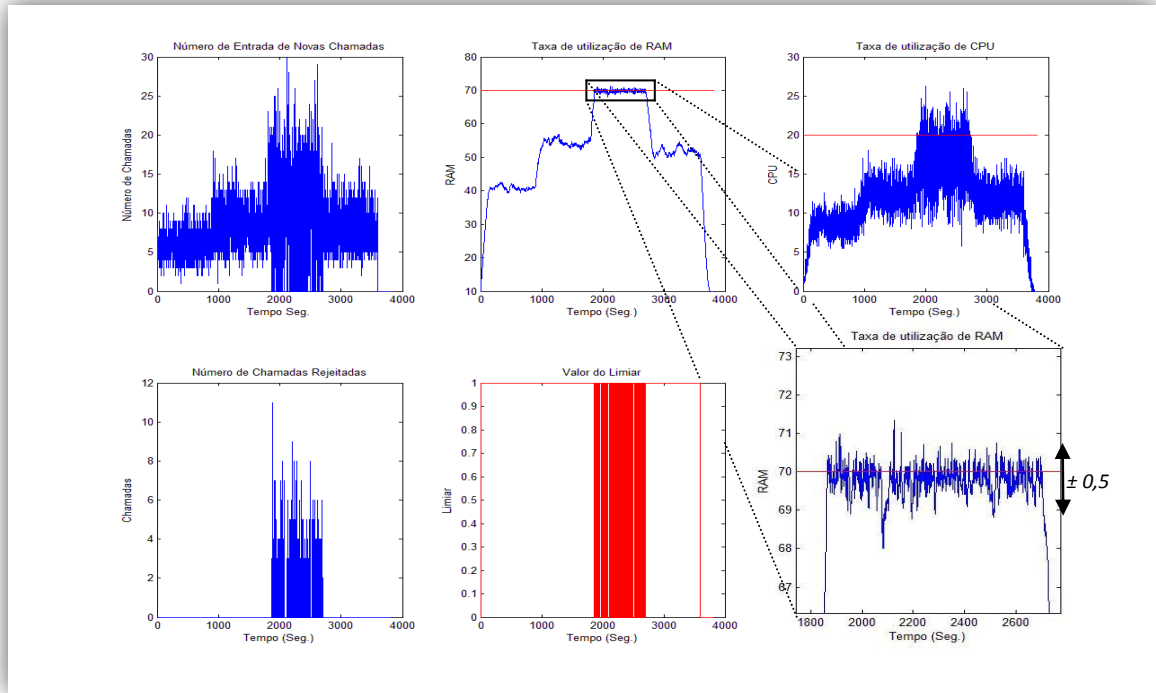


Figura 45 - Simulação do Controlo do SCF PI Janela Temporal (melhor resultado).

3.8.2 Controlo de um SCF e um SDF

Para a obtenção da largura da janela é usada a mesma malha de controlo da Figura 21, sendo a largura total da janela igual a 1 Segundo.

Controlo Proporcional:

$$Janela(n) = Control_{SCF} \times eP_{SCF}(n) \times Largura\ Total\ Janela\ (1\ Seg.) \quad (26)$$

Controlo Proporcional-Integrador:

$$Janela(n) = Control_{SCF} \times ePI_{SCF}(n) \times Largura\ Total\ Janela\ (1\ Seg.) \quad (27)$$

Em baixo podemos observar uma tabela com os resultados dos diversos controlos do SCF e SDF. Os K_p e K_I do SCF já foram calculados no ponto 3.8.1.

	K_{p_SDF}	K_{I_SDF}	Nº Chamadas Chegadas	Nº Chamadas Rejeitadas	% Chamadas Rejeitadas	EAC RAM SCF	EAC CPU SCF	EAC RAM SDF	EAC CPU SDF	EAC I/O SDF
Controlo SCF: Proporcional Controlo SDF: Proporcional	7	-----	31 506	4 651	14,7623 %	0	0,0152	0	0,140	0,0124
Controlo SCF: Proporcional Controlo SDF: PI	2	2	31 506	3 803	12,0707 %	0	0,001	$2,7733 \times 10^{-6}$	0,0638	0,0078
Controlo SCF: PI Controlo SDF: Proporcional	7	-----	31 506	4 651	14,7623 %	0	0,0152	0	0,1040	0,0124
Controlo SCF: PI Controlo SDF: PI	80	85	31 506	3 367	11,6581 %	0	0,0114	$1,0755 \times 10^{-4}$	0,0683	0,0083

Tabela 23 - Resultados do Controlo do SCF e SDF (Janelas Temporais).

De salientar que os SDF com controlo *Proporcional* apresentam os valores de rejeição mais altos. Neste caso ambas as simulações obtiveram valores iguais, o que significa que não houve influência no controlo por parte dos SCF.

A combinação *PI* no SCF e SDF tem o melhor comportamento no que respeita a rejeição de chamadas.

Os K_p e k_I foram afinados com valores compreendidos entre '0' e '100' sendo escolhido os valores que apresentavam a menor taxa de rejeição de chamadas. Durante esta afinação não foi observada qualquer situação de instabilidade.

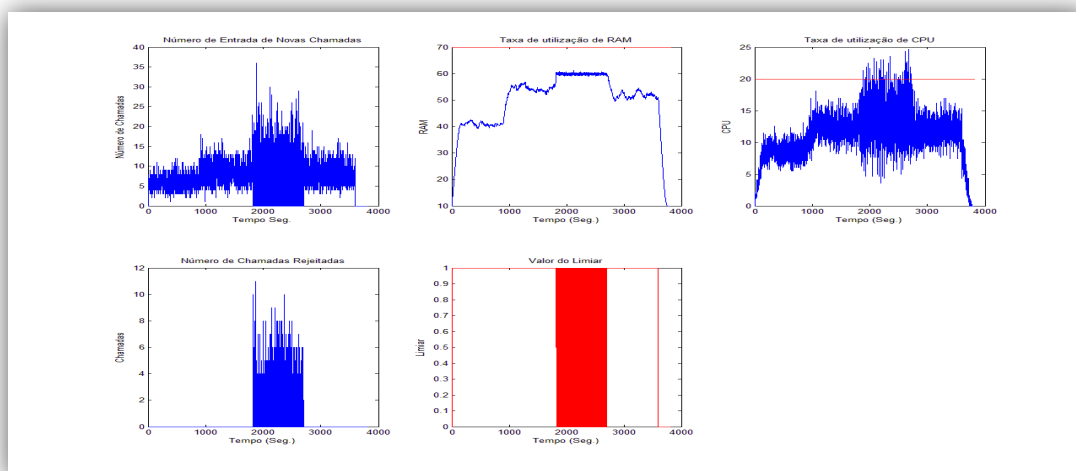


Figura 46 - SCF: Simulação do Controlo do SCF e SDF PI Janelas Temporais (melhor resultado).

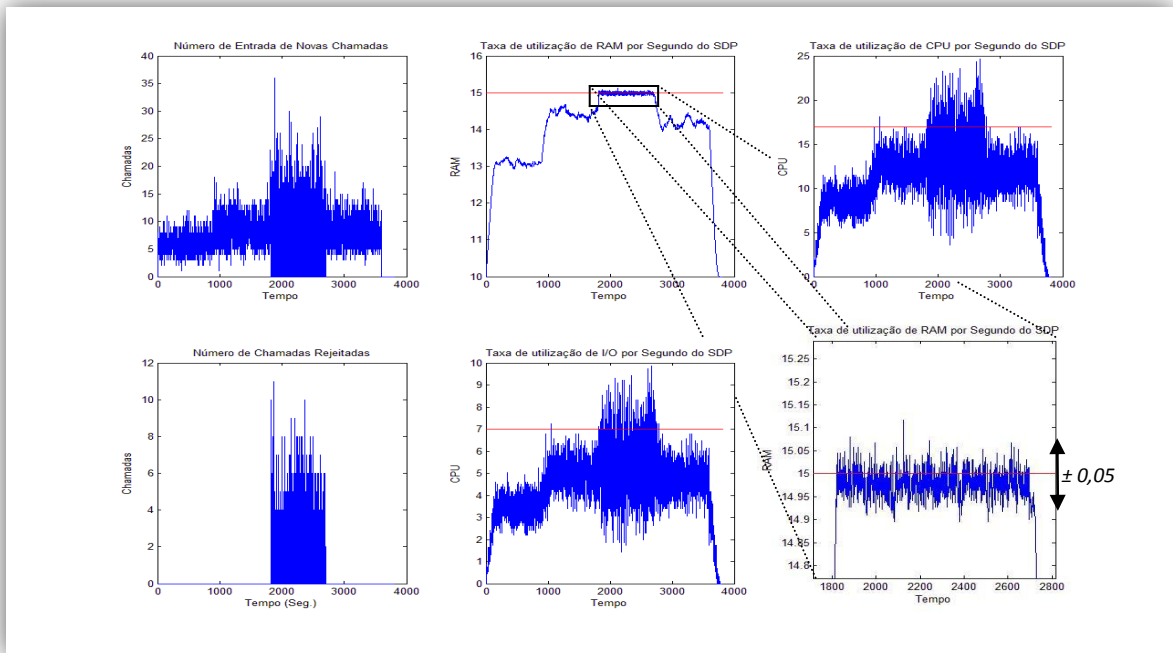


Figura 47 - SDF: Simulação do Controle do SCF e SDF PI Janelas Temporais (melhor resultado).

3.8.3 Controle de quatro SCF e um SDF

Para a obtenção da largura da janela é usada a mesma malha de controle da Figura 24 e o esquema de prioridades da Tabela 7, sendo a largura total da janela igual a 1 Segundo.

Controle Proporcional:

$$Janela_{SCF_{i,j}}(n) = \begin{cases} peso_{i,j} \cdot Control_{SCF_{i,j}}(n) \cdot eP_{SDF}(n) \cdot Largura\ Total\ Janela\ (1\ Seg.), & peso_{i,j} \neq 0 \\ Largura\ Total\ Janela\ (1\ Seg.) \cdot Control_{SCF_{i,j}}(n), & peso_{i,j} = 0 \end{cases} \quad (28)$$

i representa o número do SCF e j representa o nível em que esse SCF se encontra. Note-se que quando $peso_{i,j} = 0$ não existe influencia do SDF nos SCF.

Controle Proporcional-Integrador:

$$Janela_{SCF_{i,j}}(n) = \begin{cases} peso_{i,j} \cdot Control_{SCF_{i,j}}(n) \cdot ePI_{SDF}(n) \cdot Largura\ Total\ Janela\ (1\ Seg.), & peso_{i,j} \neq 0 \\ Largura\ Total\ Janela\ (1\ Seg.) \cdot Control_{SCF_{i,j}}(n), & peso_{i,j} = 0 \end{cases} \quad (29)$$

Todos os SCF terão sempre o mesmo tipo de controle, isto é, ou terão todos controle *Proporcional*, ou *Proporcional-Integrador*.

	K_{p_SDF}	K_{I_SDF}	Nº Chamadas Chegadas	Nº Chamadas Rejeitadas	% Chamadas Rejeitadas	EAC RAM SDF	EAC CPU SDF	EAC I/O SDF
Controlo SCF: Proporcional Controlo SDF: Proporcional	25	-----	130 500	40 988	31,4084 %	$1,0560 \times 10^{-4}$	0,4391	0,0703
Controlo SCF: Proporcional Controlo SDF: PI	5	5	130 500	35 468	27,1785 %	$4,2772 \times 10^{-4}$	0,6251	0,100
Controlo SCF: PI Controlo SDF: Proporcional	4	-----	130 500	42 295	32,4110 %	0,3311	0,0755	0,0617
Controlo SCF: PI Controlo SDF: PI	30	55	130 500	34 517	26,4498 %	0,0212	0,5721	0,0915

Tabela 24 - SDF: Resultados da Simulação com quatro SCF e um SDF (Janelas Temporais).

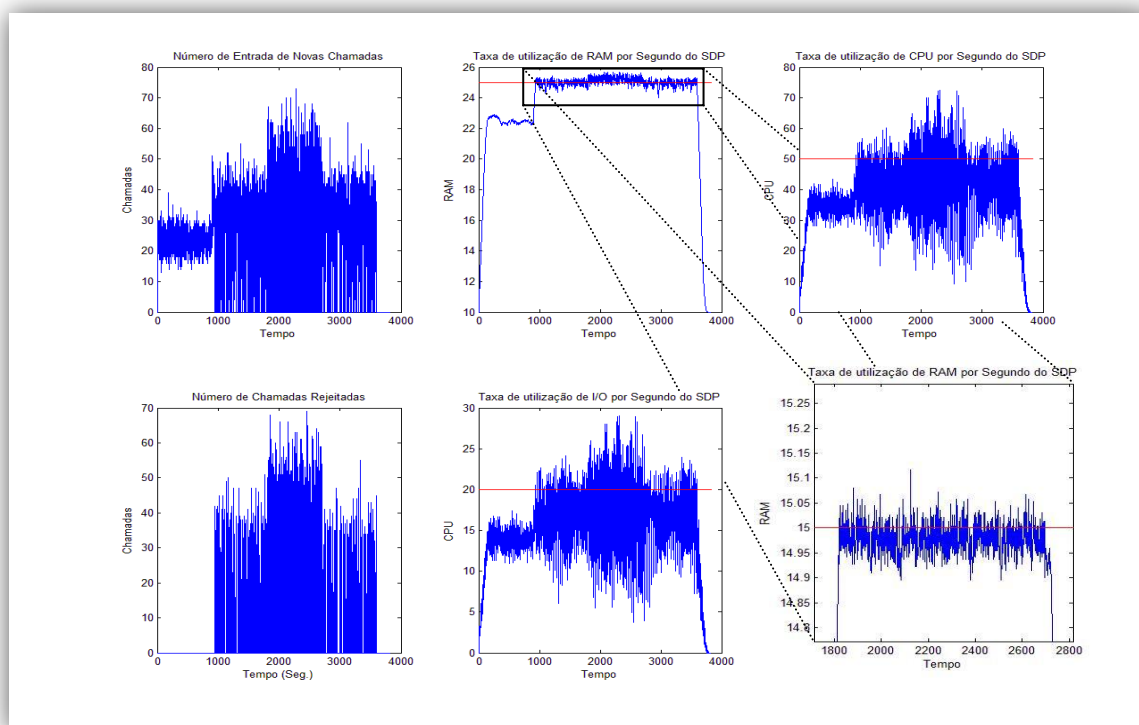


Figura 48 - SDF: Simulação com Controlo PI nos quatro SCF e no SDF Janelas Temporais (melhor resultado).

Em baixo podemos observar uma tabela com os resultados dos diversos controlos dos quatro SCF.

	SCF	K_p	K_I	Nº Chamadas Chegadas	Nº Chamadas Rejeitadas	% Chamadas Rejeitadas	EAC RAM SCF	EAC CPU SCF
Controlo SCF: Proporcional Controlo SDF: Proporcional	1º	4	---	32 505	5 198	15,9914 %	0	0,0117
	2º	4	---	32 750	9 037	27,5939 %	0	0,0012
	3º	4	---	32 757	13 292	40,5776 %	0	0
	4º	4	---	32 488	13 461	41,4338 %	0	$3,1378 \times 10^{-5}$
Controlo SCF: Proporcional Controlo SDF: PI	1º	4	---	32 505	4 368	13,4379 %	0	0,0061
	2º	4	---	32 750	7 857	23,9908 %	0	$4,3536 \times 10^{-4}$
	3º	4	---	32 757	11 620	35,4733 %	0	$3,1378 \times 10^{-5}$
	4º	4	---	32 488	11 623	35,7763 %	0	$1,9611 \times 10^{-4}$
Controlo SCF: PI Controlo SDF: Proporcional	1º	5	2	32 505	4 717	14,5116 %	0	0,0201
	2º	5	2	32 750	9 427	28,7847 %	0	$1,0394 \times 10^{-4}$
	3º	5	2	32 757	13 952	42,5924 %	0	0
	4º	5	2	32 488	14 199	43,7054 %	0	0
Controlo SCF: PI Controlo SDF: PI	1º	5	2	32 505	3 740	11,5059 %	0	0,0169
	2º	5	2	32 750	7 663	23,3985 %	0	$4,1183 \times 10^{-5}$
	3º	5	2	32 757	11 424	34,8750 %	0	$1,9611 \times 10^{-5}$
	4º	5	2	32 488	11 690	35,9825 %	0	0

Tabela 25 - SCF: Resultados da Simulação com quatro SCF e um SDF (Janelas Temporais).

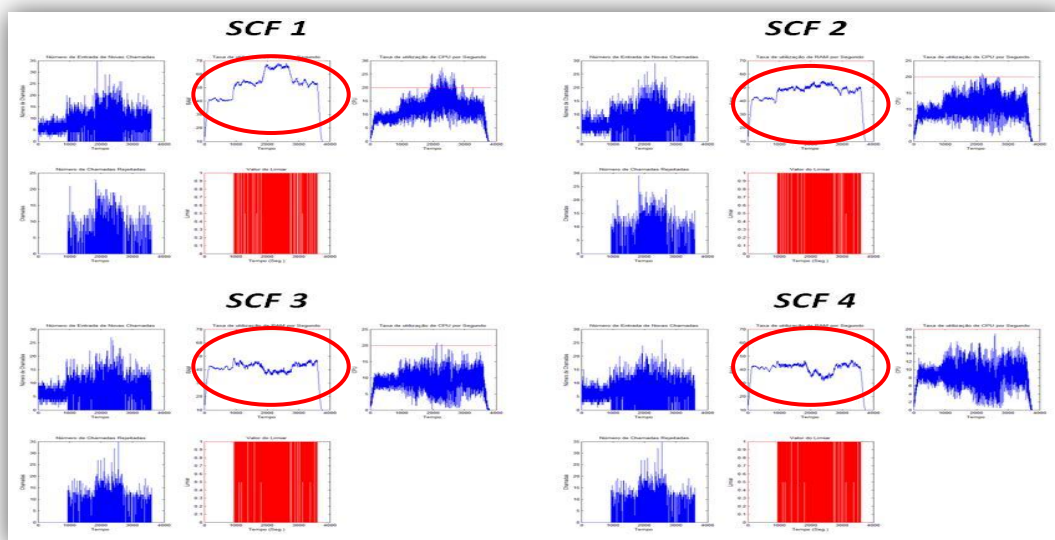


Figura 49 - SCF: Simulação com Controlo PI nos quatro SCF e no SDF Janelas Temporais (melhor resultado).

Observando a *Tabela 24* podemos dizer que a menor rejeição de chamadas acontece para a combinação *PI* no *SCF* e *SDF*. Consta-se também, que o controlo *Proporcional* no *SDF* aumenta o número de rejeição de chamadas.

Fez-se mais uma simulação aumentando para o dobro a frequência de amostragem, isto é, a duração de cada janela temporal passou a ser de 0,5 Segundos. Os dados obtidos mostram que para o caso do controlo *Proporcional* houve uma redução acentuada do número de chamadas rejeitadas, aproximando-se bastante dos valores com controlo *PI*, que não sofreram grandes alterações, baixando apenas ligeiramente. O Controlo *PI* continua a ter os valores mais baixos de rejeição de chamadas, sendo que o valor mais baixo é encontrado quando se usa controlo *PI* no *SCF* e *SDF*.

Os K_p e k_i foram afinados com valores compreendidos entre '0' e '100' sendo escolhido os valores que apresentavam a menor taxa de rejeição de chamadas. Durante esta afinação não foi observada qualquer situação de instabilidade.

3.9 Discussão de Resultados

O controlo que teve melhor comportamento a nível de rejeição de chamadas é o controlo com um *Limiar Máximo Estático* e um *Limiar mínimo Dinâmico*, sendo que o valor mais baixo do número de chamadas rejeitadas é conseguido usando controlo *Proporcional-Integrador* no *SCF* e *SDF*. Este controlo apresenta também os maiores valores de erro acumulado fruto das próprias características do controlador *PI* que permite que os limites impostos possam ser violados devido ao factor integrador, e também da própria construção do algoritmo que permite que sejam aceites mais chamadas, aumentando assim a probabilidade desses limites serem mais vezes violados.

Os erros acumulados destes controladores *PI* podem ser diminuídos bastando para isso reduzir o número de amostras de integração, mas consequentemente o número de chamadas rejeitadas irá aumentar.

O controlo com *Janelas Temporais* apresenta um grande problema, uma vez que pode acontecer que não cheguem novas chamadas na largura de tempo em que é permitida aceitação aparecendo o *burst* de chamadas na zona de rejeição, fazendo aumentar deste modo a rejeição de chamadas desnecessariamente.

Ordenando os algoritmos por ordem de menor rejeição temos as seguinte tabelas:

Um SCF		
	Controlo do Sistema	Controlo no SCF
1º	<i>Limiar Máximo Estático e Limiar mínimo Dinâmico</i>	<i>Proporcional-Integrador</i>
2º	<i>Dois Limiares Dinâmicos</i>	<i>Proporcional-Integrador</i>
3º	<i>Um Limiar Dinâmico</i>	<i>Proporcional-Integrador</i>
4º	<i>Janelas Temporais</i>	<i>Proporcional-Integrador</i>
5º	<i>Um Limiar de Eventos</i>	<i>Proporcional-Integrador</i>

Tabela 26 - Ordenação dos Algoritmos de Controlo em Relação ao menor N° de Rejeição (um SCF).

Um SCF e um SDF			
	Controlo do Sistema	Controlo no SCF	Controlo no SDF
1º	<i>Limiar Máximo Estático e Limiar mínimo Dinâmico</i>	<i>Proporcional-Integrador</i>	<i>Proporcional-Integrador</i>
2º	<i>Dois Limiares Dinâmicos</i>	<i>Proporcional-Integrador</i>	<i>Proporcional-Integrador</i>
3º	<i>Um Limiar Dinâmico</i>	<i>Proporcional-Integrador</i>	<i>Proporcional-Integrador</i>
4º	<i>Janelas Temporais</i>	<i>Proporcional-Integrador</i>	<i>Proporcional-Integrador</i>
5º	<i>Um Limiar de Eventos</i>	<i>Proporcional-Integrador</i>	<i>Proporcional-Integrador</i>

Tabela 27 - Ordenação dos Algoritmos de Controlo em Relação ao menor N° de Rejeição (um SCF e um SDF).

Quatro SCF e um SDF			
	Controlo do Sistema	Controlo no SCF	Controlo no SDF
1º	<i>Limiar Máximo Estático e Limiar mínimo Dinâmico</i>	<i>Proporcional-Integrador</i>	<i>Proporcional-Integrador</i>
2º	<i>Dois Limiares Dinâmicos</i>	<i>Proporcional-Integrador</i>	<i>Proporcional-Integrador</i>
3º	<i>Um Limiar Dinâmico</i>	<i>Proporcional-Integrador</i>	<i>Proporcional-Integrador</i>
4º	<i>Um Limiar de Eventos</i>	<i>Proporcional-Integrador</i>	<i>Proporcional-Integrador</i>
5º	<i>Janelas Temporais</i>	<i>Proporcional-Integrador</i>	<i>Proporcional-Integrador</i>

Tabela 28- Ordenação dos Algoritmos de Controlo em Relação ao menor N° de Rejeição (quatro SCF e um SDF).

De salientar que é o controlo com um *Limiar de Eventos* que apresenta os valores mais altos de rejeição de chamadas exceptuando no caso da simulação com quatro *SCF* e um *SDF* onde foi o controlo com *Janelas Temporais* que obteve pior desempenho (com cerca de mais 0,4% de rejeição de chamadas em relação ao controlo com um *Limiar de Eventos*).

Como numa implementação prática pretende-se controlar o sistema rejeitando o menor número de chamadas possível, o controlo que deverá ser implementado será o que utiliza um *Limiar Máximo Estático* e um *Limiar mínimo Dinâmico*. Basta ter um *Limiar* dinâmico calculado com controlo *Proporcional* ou *Proporcional-Integrador* para termos uma redução considerável no número de chamadas rejeitas. De notar ainda que apenas usando dois *Limiares* reduzimos bastante a rejeição de chamadas em relação ao algoritmo usando um *Limiar Estático*.

Os cálculos do k_p e k_i são difíceis de obter através de cálculos matemáticos uma vez que estamos a tentar controlar um sistema que não é linear nem invariante no tempo, mesmo assim no caso de carga máxima, e durante estas simulações, quando estas constantes não estavam afinadas provou-se ter um aumento da rejeição até 0,5%, contudo não é possível afirmar com 100% de garantias que o mesmo se passa no sistema real, o que se pode afirmar com toda a certeza é que o uso de um controlador, *Proporcional* ou *PI*, para cálculo de *Limiar Dinâmico*, reduz drasticamente o número de rejeições de chamadas.

4. Implementação

4.1 Sistema Real

A arquitectura real do sistema que se pretende controlar apresenta algumas diferenças em relação ao modelo de simulação desenvolvido. Mesmo assim espera-se obter as mesmas conclusões visto que o funcionamento geral do algoritmo de controlo não foi alterado.

Apenas foi implementado parte do algoritmo de controlo no *SCF* ficando para implementação futura a totalidade do algoritmo de controlo no *SCF* e *SDF*.

A arquitectura do sistema real onde se implementou parte do algoritmo de controlo é do tipo *Cliente-Servidor*, sendo o *SCF* o cliente e o *SDF* o servidor, e pode ser descrita pelo seguinte diagrama:

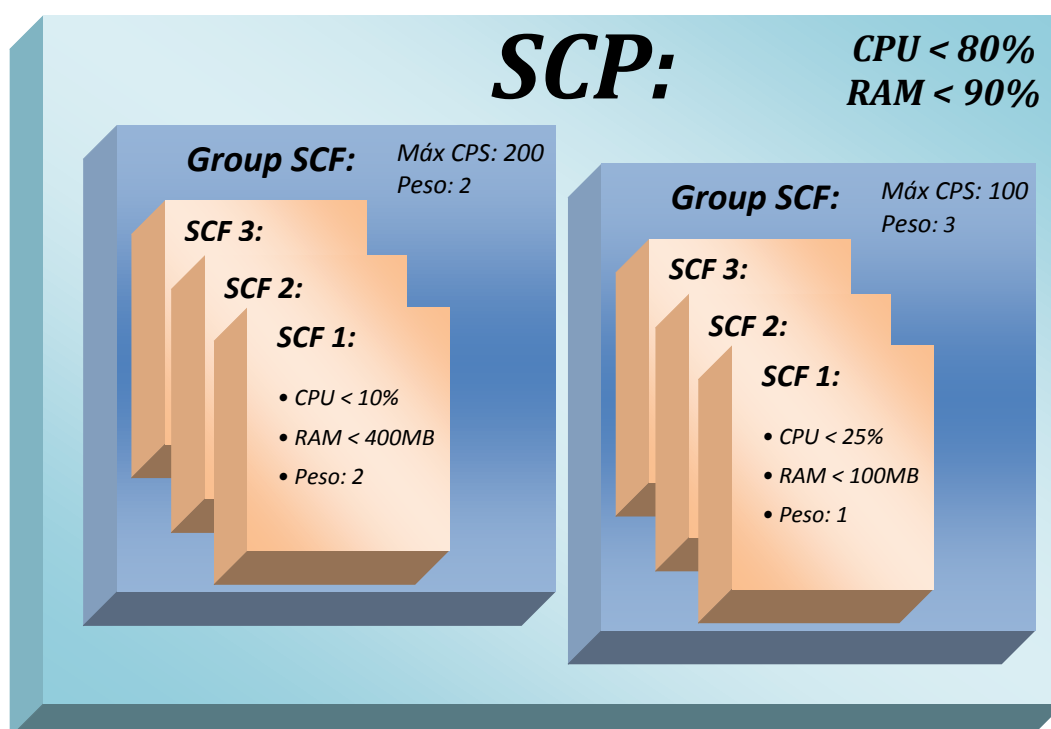


Figura 50 – Arquitectura do Sistema Real.

Uma Máquina *SCP* pode conter vários grupos de *SCF*. Cada grupo de *SCF* pode corresponder a um tipo de cliente, por exemplo, os clientes de uma rede móvel com cartões pré-pagos, e cada *SCF* corresponde a um tipo de serviço, como por exemplo *MMS* ou *SMS*.

O objectivo do controlo é assegurar qualidade de serviço e respeitar as condições impostas pelos clientes.

Como podemos ver pela *Figura 51* a malha de controlo é muito semelhante à malha de controlo com *SDP* e *SCF* que encontramos na *Figura 21* o que motiva a comparação com os resultados obtidos nas simulações de controlo visto que o funcionamento geral da malha de controlo e respectivas fórmulas não foram alteradas, apenas adaptadas.

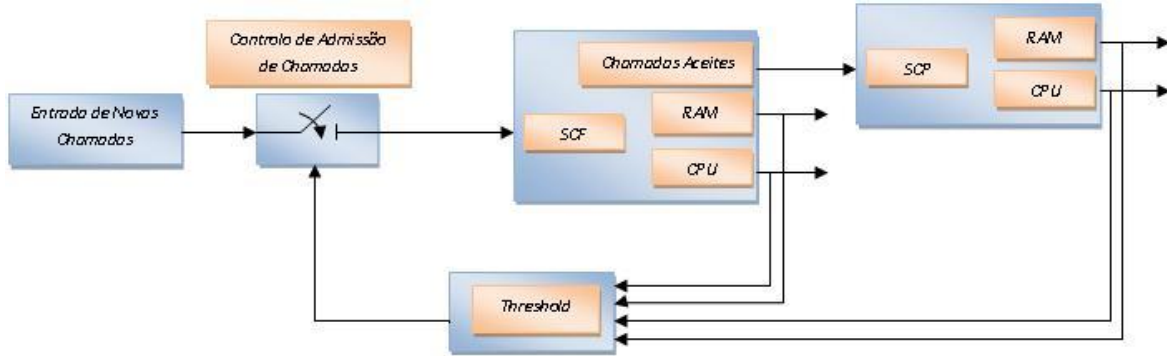


Figura 51 - Malha de Controlo do Sistema Real.

Como podemos observar existe uma diferença ligeira da malha de controlo da *Figura 21*. A malha da *Figura 21* tem mais um parâmetro de controlo, o *I/O*, que não vai ser introduzido neste caso para já, visto que por enquanto não existe modo de obter essa medida. Após os estudos teóricos foi concluído que os melhores resultados são obtidos usando controlo com *um Limiar Máximo Estático e um Limiar mínimo Dinâmico* com controlador *PI*, logo é esse o controlo que foi implementado no sistema real.

Para o cálculo do *Limiar mínimo Dinâmico* foram usadas as fórmulas (11) a (14) para o controlo do *SCF* e as fórmulas (30) e (18) a (20) para o controlo do *SCF* com *SDF*:

$$eT_{SDF}(n) = (Y^T - Y(n)) \Big|_0^1 \cdot (\beta^T - \beta(n)) \Big|_0^1 \quad (30)$$

Cada Grupo de *SCF* tem que respeitar um valor máximo imposto de *CPS*, calculado através de pesos que cada grupo tem no *SCP*. Estas *Calls per second* são distribuídas pelos *SCF* de cada grupo equitativamente ou consoante um determinado peso. Os pesos vão de '1' a '10', sendo '10' o maior peso e '1' o menor. O algoritmo de controlo não permite que a soma dos pesos seja superior a '10'. O *Limiar Máximo* é obtido então com as seguintes fórmulas:

$$Limiar\ Grupo(n) = \frac{Máximo\ Calls\ per\ Second\ do\ SCP}{n^o\ de\ Grupos\ de\ SCFs} \quad (31)$$

Para o caso de uma distribuição equitativa do *Limiar*:

$$\text{Limiar Máximo}(n) = \frac{\text{Limiar Grupo}}{n^{\circ} \text{ de SCFs}} \quad (32)$$

Para o caso de serem atribuídos pesos aos SCF:

$$\text{Limiar Máximo}(n) = \frac{\text{Peso}}{10} \cdot \text{Limiar Grupo} \quad (33)$$

Como se pode, ver as fórmulas de controlo do sistema real são em tudo semelhantes às fórmulas do estudo teórico apresentando ligeiras adaptações. Uma das adaptações diz respeito à forma como são actualizados os valores instantâneos de *RAM* e *CPU*. No modelo teórico os dados de *RAM* e *CPU* são actualizados em cada instante, isto é em cada 0,1 Segundos, o que nos permite ter também um cálculo do *Limiar* instantâneo. No modelo real por requisitos do sistema, os dados de *RAM* e *CPU* apenas nos podem ser facultados em cada 1 Segundo reflectindo-se isso no cálculo do *Limiar*.

Como já foi referido, o sistema real tem uma arquitectura do tipo *cliente-servidor* e por isso foi criado um algoritmo de controlo para o cliente (*SCF*) e outro para o servidor (*SDF*). O algoritmo de controlo do cliente foi implementado na linguagem *C/C++* sendo depois convertido para *HP-UX* com criação das respectivas bibliotecas e introduzido no sistema. O algoritmo de controlo do servidor encontra-se apenas em modelo de testes simulado na linguagem *C/C++*, não tendo sido ainda introduzido no sistema por razões de ordem técnica.

4.1 Controlo de Carga no SCF (Cliente)

Para o controlo de carga no cliente foi criada e introduzida no sistema a classe de objectos *LoadControlClientCls*. Esta classe implementa no sistema o algoritmo de controlo com dois *Limiares Estáticos*. Como já foi referido no capítulo 3, este controlo impõe um limite máximo de *Calls per Second* e um limite mínimo que pode ser violado uma vez antes de dar origem à rejeição de chamadas (*Figura 27*).

O *Limiar Máximo* e o *Limiar mínimo* são definidos por defeito com os valores 15 CPS e 10 CPS respectivamente. Estes valores podem ser alterados com a função *setControlValues*.

O controlo é feito através da função *getPermission*. Sempre que chega uma nova chamada é invocada esta função que devolve '1' no caso da chamada poder ser aceite ou '0' se a chamada tiver que ser rejeitada. Esta função tem um contador de chamadas, o qual é posto a '0' no início de cada segundo, que conta o número de chamadas aceites em

cada segundo e faz a comparação deste valor com os *Limiares* definidos. Inicialmente este contador não existia porque o sistema possui um método que nos devolve as *CPS*, infelizmente este método funciona mal porque utiliza relógios diferentes para a sua sincronização devolvendo-nos a taxa errada de *CPS* o que nos obrigou a criar o contador de chamadas.

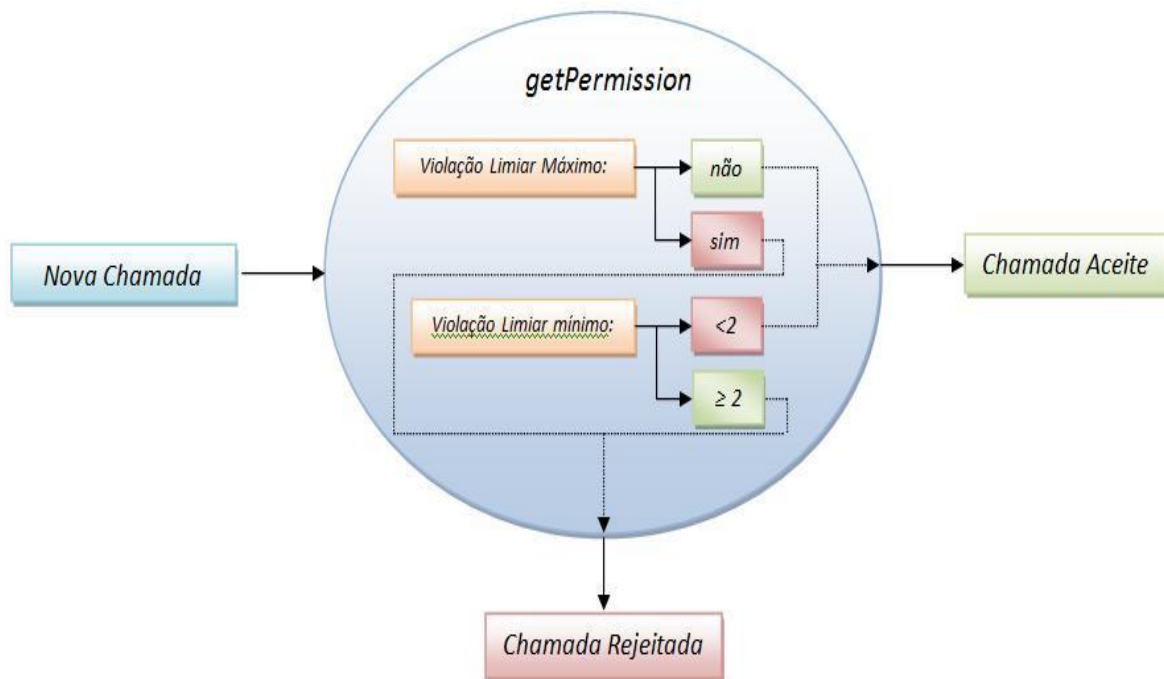


Figura 52 - Diagrama de Funcionamento de função *getPermission*.

A função *getPermission* possui ainda um contador de violações do *Limiar* mínimo que é colocado a '0' sempre que houver a segunda violação ou tiver passado mais de 2 Segundos desde a 1ª violação.

Para a introdução deste algoritmo de controlo no sistema foi imposto o requisito de que tinha que ser *ThreadSafe* por isso foi incluída a livreria "*pthread.h*".

Foram efectuados testes numa zona estanque do sistema e própria para testes para verificar se o controlo estava a funcionar correctamente. Definiu-se o *Limiar Máximo* com 50 *CPS* e o *Limiar mínimo* com 10 *CPS* e foram geradas 20 *CPS* durante 100 Segundos com a ajuda de um gerador de chamadas de teste próprio do sistema.

```
10.112.48.214 - PuTTY
2008-06-05 16:03:38.147 SCF_SDL INF 4 | CCF:StartCall - SIBEng of CallID 230 is OK
2008-06-05 16:03:38.147 SCF_SDL INF 5 | SLEF:StartCall(231) - start OK
2008-06-05 16:03:38.147 SCF_SDL INF 4 | CCF:StartCall - SIBEng of CallID 231 is OK
2008-06-05 16:03:38.148 SCF_SDL INF 5 | SLEF:StartCall(232) - start OK
2008-06-05 16:03:38.148 SCF_SDL INF 4 | CCF:StartCall - SIBEng of CallID 232 is OK
2008-06-05 16:03:38.148 SCF_SDL INF 5 | SLEF:StartCall(233) - start OK
2008-06-05 16:03:38.148 SCF_SDL INF 4 | CCF:StartCall - SIBEng of CallID 233 is OK
2008-06-05 16:03:38.148 SCF_SDL INF 5 | SLEF:StartCall(234) - start OK
2008-06-05 16:03:38.148 SCF_SDL INF 4 | CCF:StartCall - SIBEng of CallID 234 is OK
2008-06-05 16:03:38.148 SCF_SDL INF 5 | SLEF:StartCall(235) - start OK
2008-06-05 16:03:38.148 SCF_SDL INF 4 | CCF:StartCall - SIBEng of CallID 235 is OK
2008-06-05 16:03:38.148 SCF_SDL INF 5 | SLEF:StartCall(236) - start OK
2008-06-05 16:03:38.148 SCF_SDL INF 4 | CCF:StartCall - SIBEng of CallID 236 is OK
2008-06-05 16:03:38.148 SCF_SDL INF 5 | SLEF:StartCall(237) - start OK
2008-06-05 16:03:38.148 SCF_SDL INF 4 | CCF:StartCall - SIBEng of CallID 237 is OK
2008-06-05 16:03:38.148 SCF_SDL INF 5 | SLEF:StartCall(238) - start OK
2008-06-05 16:03:38.148 SCF_SDL INF 4 | CCF:StartCall - SIBEng of CallID 238 is OK
2008-06-05 16:03:38.148 SCF_SDL INF 5 | SLEF:StartCall(239) - start OK
2008-06-05 16:03:38.148 SCF_SDL INF 4 | CCF:StartCall - SIBEng of CallID 239 is OK
2008-06-05 16:03:38.148 SCF_SDL INF 5 | SLEF:StartCall(240) - start OK
2008-06-05 16:03:38.148 SCF_SDL INF 4 | CCF:StartCall - SIBEng of CallID 240 is OK
2008-06-05 16:03:38.148 SCF_SDL INF 5 | SLEF:StartCall(241) - start OK
2008-06-05 16:03:38.148 SCF_SDL INF 4 | CCF:StartCall - SIBEng of CallID 241 is OK
2008-06-05 16:03:38.148 SCF_SDL INF 5 | SLEF:StartCall(242) - start OK
2008-06-05 16:03:38.148 SCF_SDL INF 4 | CCF:StartCall - SIBEng of CallID 242 is OK
2008-06-05 16:03:38.148 SCF_SDL INF 5 | SLEF:StartCall(243) - start OK
2008-06-05 16:03:38.148 SCF_SDL INF 4 | CCF:StartCall - SIBEng of CallID 243 is OK
2008-06-05 16:03:38.148 SCF_SDL INF 5 | SLEF:StartCall(244) - start OK
2008-06-05 16:03:38.148 SCF_SDL INF 4 | CCF:StartCall - SIBEng of CallID 244 is OK
2008-06-05 16:03:38.148 SCF_SDL INF 5 | SLEF:StartCall(245) - start OK
2008-06-05 16:03:38.148 SCF_SDL INF 4 | CCF:StartCall - SIBEng of CallID 245 is OK
2008-06-05 16:03:38.148 SCF_SDL INF 5 | SLEF:StartCall(246) - start OK
2008-06-05 16:03:38.148 SCF_SDL INF 4 | CCF:StartCall - SIBEng of CallID 246 is OK
2008-06-05 16:03:38.148 SCF_SDL INF 5 | SLEF:StartCall(247) - start OK
2008-06-05 16:03:38.148 SCF_SDL INF 4 | CCF:StartCall - SIBEng of CallID 247 is OK
2008-06-05 16:03:38.148 SCF_SDL INF 5 | SLEF:StartCall(248) - start OK
2008-06-05 16:03:38.148 SCF_SDL INF 4 | CCF:StartCall - SIBEng of CallID 248 is OK
2008-06-05 16:03:38.148 SCF_SDL INF 5 | SLEF:StartCall(249) - start OK
2008-06-05 16:03:38.148 SCF_SDL INF 4 | CCF:StartCall - SIBEng of CallID 249 is OK
2008-06-05 16:03:38.148 SCF_SDL INF 5 | SLEF:StartCall(250) - start OK
2008-06-05 16:03:38.148 SCF_SDL INF 4 | CCF:StartCall - SIBEng of CallID 250 is OK
2008-06-05 16:03:38.148 SCF_ENV INF 1 | Starting TCAP in INACTIVE mode
```

Figura 53- Arranque do SCF.

A Figura 53 mostra o bom arranque do SCF, uma vez que não foram emitidas mensagens de erro, pelo que podemos considerar os resultados dos testes válidos.

```
2008-06-05 16:24:35.723 SCF_SDL WRN 3 | Call: 239 service: 30 with error in SIB: 2 for Clg:
2008-06-05 16:24:35.723 SCF_SIBCS1V2 WRN 5 | Cause is out of limits! 0 not in [1,127]. ReleaseCall will be sent without cause!
2008-06-05 16:24:35.724 SCF_SDL ERR 4 | Retriggerring failed because of undefined Service: 55
2008-06-05 16:24:35.724 SCF_SIB WRN 1 | SDMSIB: call 240 and SIBNum=2 : Error trying to retrigger to another SK...
2008-06-05 16:24:35.724 SCF_SDL WRN 3 | Call: 240 service: 30 with error in SIB: 2 for Clg:
2008-06-05 16:24:35.724 SCF_SIBCS1V2 WRN 5 | Cause is out of limits! 0 not in [1,127]. ReleaseCall will be sent without cause!
2008-06-05 16:24:36.203 SCF_ENV WRN 5 | Call rejected by CPS overload [In: 11, Max: 10]
2008-06-05 16:24:36.203 SCF_ENV WRN 5 | Call rejected by CPS overload [In: 12, Max: 10]
2008-06-05 16:24:36.203 SCF_ENV WRN 5 | Call rejected by CPS overload [In: 13, Max: 10]
2008-06-05 16:24:36.203 SCF_ENV WRN 5 | Call rejected by CPS overload [In: 14, Max: 10]
2008-06-05 16:24:36.203 SCF_ENV WRN 5 | Call rejected by CPS overload [In: 15, Max: 10]
2008-06-05 16:24:36.203 SCF_ENV WRN 5 | Call rejected by CPS overload [In: 16, Max: 10]
2008-06-05 16:24:36.203 SCF_ENV WRN 5 | Call rejected by CPS overload [In: 17, Max: 10]
2008-06-05 16:24:36.203 SCF_ENV WRN 5 | Call rejected by CPS overload [In: 18, Max: 10]
2008-06-05 16:24:36.203 SCF_ENV WRN 5 | Call rejected by CPS overload [In: 19, Max: 10]
2008-06-05 16:24:36.204 SCF_ENV WRN 5 | Call rejected by CPS overload [In: 20, Max: 10]
2008-06-05 16:24:36.720 SCF_SDL ERR 4 | Retriggerring failed because of undefined Service: 55
2008-06-05 16:24:36.720 SCF_SIB WRN 1 | SDMSIB: call 241 and SIBNum=2 : Error trying to retrigger to another SK...
2008-06-05 16:24:36.720 SCF_SDL WRN 3 | Call: 241 service: 30 with error in SIB: 2 for Clg:
2008-06-05 16:24:36.721 SCF_SIBCS1V2 WRN 5 | Cause is out of limits! 0 not in [1,127]. ReleaseCall will be sent without cause!
2008-06-05 16:24:36.721 SCF_SDL ERR 4 | Retriggerring failed because of undefined Service: 55
2008-06-05 16:24:36.721 SCF_SIB WRN 1 | SDMSIB: call 242 and SIBNum=2 : Error trying to retrigger to another SK...
2008-06-05 16:24:36.721 SCF_SDL WRN 3 | Call: 242 service: 30 with error in SIB: 2 for Clg:
2008-06-05 16:24:36.721 SCF_SIBCS1V2 WRN 5 | Cause is out of limits! 0 not in [1,127]. ReleaseCall will be sent without cause!
2008-06-05 16:24:36.721 SCF_SDL ERR 4 | Retriggerring failed because of undefined Service: 55
```

Figura 54 - Log doSCF.

Pela Figura 54 conseguimos observar que há uma 2ª violação do Limiar mínimo e dando origem a rejeição de chamadas.


```

FINAL REPORT

Begin Time :    2008-06-05 16:22:55
End Time   :    2008-06-05 16:24:38
Total      :    +102.385343 sec

Total number of calls :    2000
Success calls :    1500 ( 75%)
Error calls :    500 ( 25%)
  IDP Timeout :    0
  ACR Timeout :    0
  EVR Timeout :    0
  Other errors :    500
Averages for success calls:
  Connect Time :    0.400270 (Max : 0.561659)
  Answer Time  :    0.409404 (Max : 0.586850)
  Disconnect Time : 0.000000 (Max : 0.000000)
  End Time     :    0.596018 (Max : 0.831925)

Closing SDL-system environment
kapa:msantos /ngin/users/msantos/Marco/SCF_SRC/WorkDir/TestZone/bin>

```

Figura 55 - Resultados dos testes do Algoritmo de Controlo no SCF.

São geradas durante 100 Segundos exactamente 20 CPS, o que implica que haja uma segunda violação a cada 2 Segundos. Como consequência, haverá apenas a aceitação de 10 CPS nestes segundos. Será de esperar portanto, aceitação de 20 CPS em 50 Segundos da simulação total e aceitação de apenas 10 CPS nos outros 50 Segundos:

$$\text{Total Chamadas} = 100 \text{ Segundos} \times 20 \text{ CPS} = 2000 \text{ Chamadas} \quad (34)$$

$$\text{Chamadas Aceites} = 50 \text{ Segundos} \times 20 \text{ CPS} = 1500 \text{ Chamadas} \quad (35)$$

$$\text{Chamadas Rejeitadas} = 50 \text{ Segundos} \times 10 \text{ CPS} = 500 \text{ Chamadas} \quad (36)$$

A Figura 55 mostra o resultado dos testes feitos ao algoritmo de controlo implementado no SCF. Nestes testes obtiveram-se 1500 chamadas aceites (75%) e 500 chamadas rejeitadas (25%) confirmando assim o bom funcionamento do algoritmo de controlo.

4.2 Controlo de Carga no SDF (Servidor)

É no servidor que vai ser introduzido o algoritmo de controlo para o cálculo dinâmico do *Limiar mínimo*. Este algoritmo encontra-se ainda numa versão de testes simulada, porque ainda não nos é possível uma obtenção dos valores de *RAM* e *CPU* do sistema com uma taxa de actualização de 1 Segundo. Esta versão de teste simula o crescimento e a diminuição de *RAM* e *CPU* do sistema para que depois seja possível o cálculo do *Limiar Dinâmico*. Pretende-se nesta fase apenas observar o correcto funcionamento do algoritmo de controlo.

Para poder implementar este controlo foram criadas 3 classes de objectos, *LoadControlServerCls*, *LoadControlGroupCls* e *LoadControlServiceCls*. Foi criada ainda mais uma classe de objectos, *System*, que tem apenas como função simular a *RAM* e o *CPU* do sistema.

A classe *System* tem duas funções, a *inc(float RAM, float CPU)* e a função *getInformation(void)*. A função *inc* simula a *RAM* e o *CPU* que incrementando-os com os factores que são inseridos como parâmetros nesta função até ser atingida uma taxa de ocupação de 81%. Após ser atingida a taxa de ocupação de 81% os valores de *RAM* e *CPU* são decrementados com os mesmos factores da incrementação até ser atingido o valor '0' que uma vez atingido dá origem novamente à incrementação.

A função *getInformation* devolve os actuais valores de *RAM* e *CPU*.

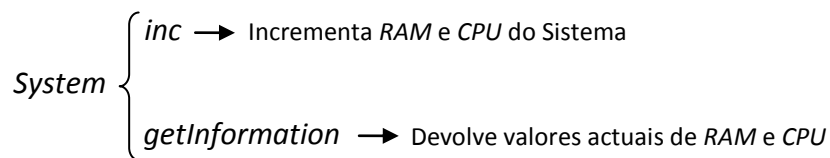


Figura 56 - Diagrama de Funções da Classe *System*.

A classe *LoadControlServerCls* define os valores limites pretendidos para a *RAM* e o *CPU* do *SCP*. Tem como função ainda adicionar novos grupos de *SCF* e guardar informações sobre esses grupos. Nesta classe é definido também se os grupos de *SCF* têm pesos diferentes ou se têm pesos iguais.

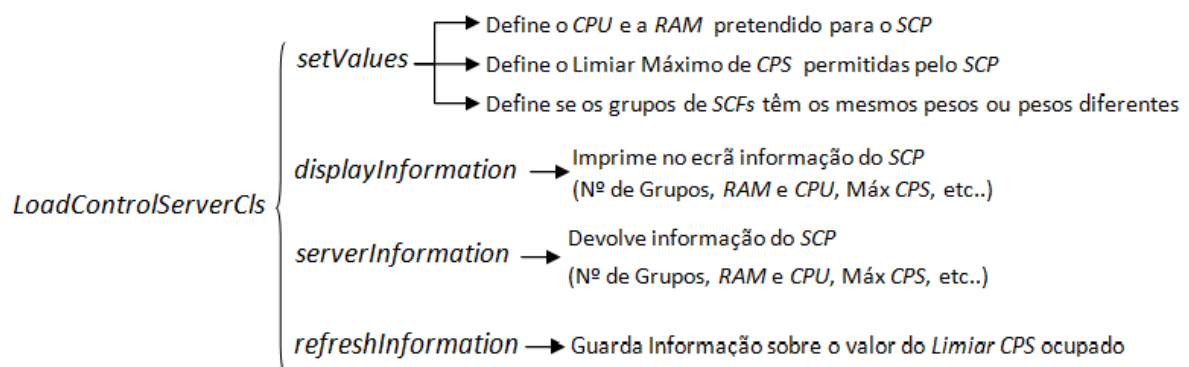


Figura 57 - Diagrama de Funções da Classe *LoadControlServerCls*.

A classe *LoadControlGroupCls* tem como objectivo calcular o valor do *Limiar* mínimo para cada grupo de *SCF* consoante as taxas de ocupação do sistema e através de um controlo *PI*. Através da função *setValues* podemos definir o *Limiar* de cada grupo de *SCF* ou então os pesos que se pretendem. Nesta classe também são definidos o número de *SCF* de cada grupo assim como se os *SCF* têm pesos diferentes ou iguais. Para testes foi acrescentada a função *inc(float RAM, float CPU)* que serve para ir incrementando e decrementando a *RAM* e o *CPU* de cada grupo de *SCF*.

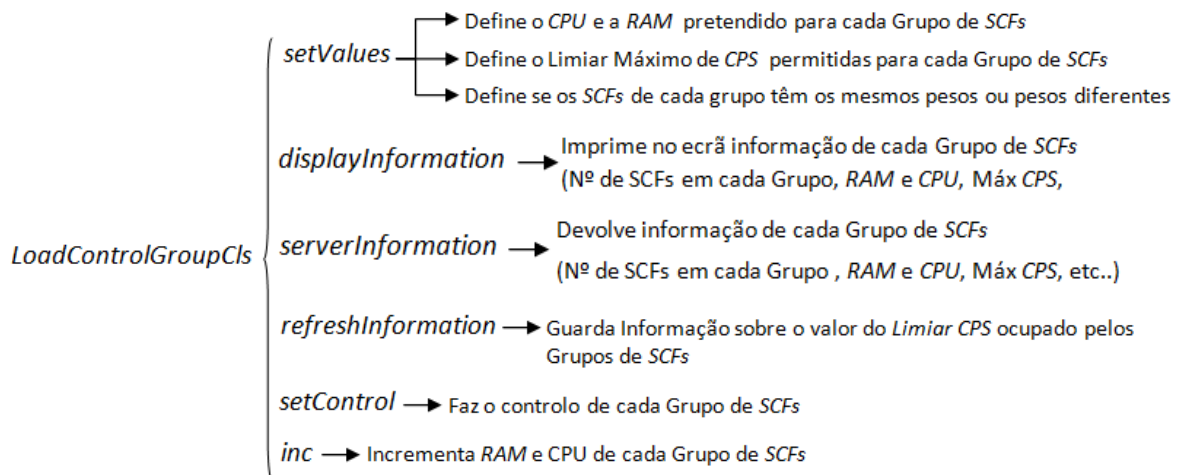


Figura 58 - Diagrama de Funções da Classe *LoadControlGroupCls*.

A classe *LoadControlServiceCls* tem como objectivo controlar cada *SCF* calculando o valor do *Limiar* mínimo consoante as taxas de ocupação do sistema e de modo a conseguir respeitar as condições de serviço impostas. Nesta classe são definidas a taxa de ocupação que cada *SCF* pode ter no sistema e o respectivo peso de cada um.

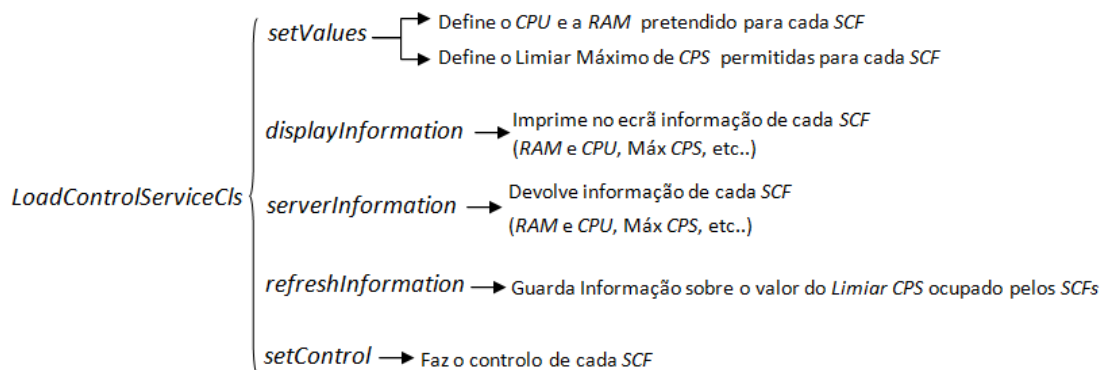


Figura 59 - Diagrama de Funções da Classe *LoadControlServiceCls*.

Durante a criação do algoritmo de controlo foram detectadas alguns problemas. Um deles é referente ao cálculo da *Fórmula (11)*. Na simulação obtínhamos valores de *RAM* e *CPU* à décima, o que dava uma margem de 10 unidades para fazer actuar o controlo. No modelo físico real, estes valores são à unidade, o que obrigou à colocação de um factor multiplicativo inferior a 1 na *Fórmula (11)*, para que deste modo se pudesse estender a gama de controlo. Os k_p e os k_i foram definidos com valores inferiores à unidade escolhidos ao acaso, 0,4 e 0,2 respectivamente, não estando por isso o controlador *Proporcional-Integrador* devidamente afinado (relembrar que nesta fase apenas se pretende observar o correcto funcionamento do algoritmo de controlo, isto é a variação, dos *Limiares Dinâmicos*). Verificamos ainda, que com estes valores não ocorre tanta saturação do *Limiar*, como aconteceu durante os testes teóricos, apresentando uma variação mais suave.

Para observar o bom funcionamento destas classes foi feito um teste onde foram criados 2 grupos de *SCF*. O 1º grupo é constituído por 2 *SCF* com pesos iguais e com limites de *CPU* e *RAM* de 20% e 30% respectivamente, e o 2º grupo por 3 *SCF* com pesos de 5, 3 e 2 com limites de *CPU* e *RAM* de 20%, 30% e 25% respectivamente. Para o 1º grupo foi definido um *Limiar Máximo* de 500 *CPS* e para o 2º, um *Limiar Máximo* de 400 *CPS*. O *SCP* tem um limite Máximo de carga de 80% para a *RAM* e para o *CPU*.

=====									
MAX CPU: 80		Real CPU: 72		MAX RAM: 80		Real RAM: 72		N. Groups: 2	
Threshold: 1000		Total Threshold: 0						Type: v	
=====									
CPU: 22.25		RAM: 22.25		Threshold: 500		Act. Threshold: 500		Total Threshold: 0	
: 0		N. Servicios: 2		Control Type: v					
CPU: 72		RAM: 72		eCPU: 1		eRAM: 1		Threshold: 500	
eP: 1		eI: 5.2		ePI: 1					
=====									
CPU: 22.5		RAM: 22.5		Threshold: 500		Act. Threshold: 500		Total Threshold: 0	
: 0		N. Servicios: 2		Control Type: v					
CPU: 20		RAM: 20		Act. CPU: 22.5		Act. RAM: 22.5		Threshold: 1	
CPU: 30		RAM: 30		Act. CPU: 22.5		Act. RAM: 22.5		Threshold: 250	
								} 2 SCF	
=====									
CPU: 22.25		RAM: 22.25		Threshold: 400		Act. Threshold: 400		Total Threshold: 0	
: 0		N. Servicios: 3		Control Type: v					
CPU: 72		RAM: 72		eCPU: 1		eRAM: 1		Threshold: 400	
eP: 1		eI: 5.2		ePI: 1					
=====									
CPU: 22.5		RAM: 22.5		Threshold: 400		Act. Threshold: 400		Total Threshold: 0	
: 0		N. Servicios: 3		Control Type: v					
CPU: 20		RAM: 20		Act. CPU: 22.5		Act. RAM: 22.5		Threshold: 1	
CPU: 30		RAM: 30		Act. CPU: 22.5		Act. RAM: 22.5		Threshold: 120	
CPU: 25		RAM: 25		Act. CPU: 22.5		Act. RAM: 22.5		Threshold: 80	
								} 3 SCF	
=====									

Figura 60 - Teste Simulado do Controlo no Servidor no SCF.

Na *Figura 60* podemos observar dentro de cada rectângulos vermelho informação relativa a cada *Grupo* de *SCF* e dentro de cada rectângulo verde informação relativa aos *SCF* de cada *Grupo*. Dentro do rectângulo azul encontra-se informação do *SCP*. Note-se que a palavra *Threshold* significa *Limiar*.

```

=====
MAX CPU: 80   Real CPU: 75   MAX RAM: 80   Real RAM: 75   N. Groups: 2   Type: v
Threshold: 1000   Total Threshold: 0

CPU: 21.5   RAM: 21.5   Threshold: 500   Act. Threshold: 424   Total Threshold:
0   N. Servicios: 2   Control Type: v
CPU: 75   RAM: 75   eCPU: 1   eRAM: 1   Threshold: 424   eP: 1   eI: 2.24   ePI:
0.848

CPU: 21.75   RAM: 21.75   Threshold: 500   Act. Threshold: 424   Total Threshold
: 0   N. Servicios: 2   Control Type: v
CPU: 20   RAM: 20   Act. CPU: 21.75   Act. RAM: 21.75   Threshold: 1
CPU: 30   RAM: 30   Act. CPU: 21.75   Act. RAM: 21.75   Threshold: 212

CPU: 21.5   RAM: 21.5   Threshold: 400   Act. Threshold: 339.2   Total Threshold
: 0   N. Servicios: 3   Control Type: v
CPU: 75   RAM: 75   eCPU: 1   eRAM: 1   Threshold: 339.2   eP: 1   eI: 2.24   eP
I: 0.848

CPU: 21.75   RAM: 21.75   Threshold: 400   Act. Threshold: 339.2   Total Thresho
ld: 0   N. Servicios: 3   Control Type: v
CPU: 20   RAM: 20   Act. CPU: 21.75   Act. RAM: 21.75   Threshold: 1
CPU: 30   RAM: 30   Act. CPU: 21.75   Act. RAM: 21.75   Threshold: 101.76
CPU: 25   RAM: 25   Act. CPU: 21.75   Act. RAM: 21.75   Threshold: 67.84
=====

```

Figura 61 - Teste Simulado do Controlo no Servidor no SCF.

Na Figura 61 podemos observar dentro dos círculos verdes os *Limiares Máximos* definidos e dentro dos círculos vermelhos a variação desses *Limiares* devido ao funcionamento do algoritmo de controlo de carga. Nestas condições podemos afirmar que o algoritmo de controlo funciona adequadamente uma vez que se verifica a variação dos *Limiares* dos Grupos de SCF consoante a carga (RAM e CPU) do sistema e a variação dos *Limiares* de cada SCF consoante a carga do respectivo Grupo.

5. Conclusão e Trabalho Futuro

5.1 Conclusão

O conceito de “Redes Inteligentes” (*Intelligent Networks*) corresponde a uma nova arquitectura de redes que surgiu com a crescente procura de novos serviços e da maior flexibilidade do serviço telefónico. Esta arquitectura assenta na separação do suporte lógico do serviço por um lado, e das funções de processamento e transporte da chamada por outro. Deste modo, a introdução de serviços suplementares, o seu controlo e a sua gestão é conseguida de um modo mais rápido, eficaz e económico.

Estas redes são compostas, numa simplificada, por *SCP* (*Service Control Points*) e *SDP* (*Service Data Points*). Os *SCP* fornecem a Lógica de Serviço das Redes Inteligentes e os *SDP* contêm os dados referentes a essa lógica de Serviços. Um *SDP* pode conter os dados de vários *SCF*. Estes nós são vitais para as Redes Inteligentes e podem ser acedidos por vários utilizadores situados em pontos diferentes. Numa arquitectura distribuída, como é o caso desta rede, onde cada utilizador actua de forma independente, poderemos encontrar facilmente situações de sobrecarga que podem criar o colapso de toda a Rede. A reconfiguração dos sistemas de suporte às redes inteligentes vai permitir um melhor aproveitamento dos recursos físicos do sistema rejeitando o mínimo de chamadas para assegurar estabilidade e bom desempenho.

Nesta dissertação foram estudadas e desenvolvidas metodologias de controlo que promovem a reconfiguração automática destas redes e que assegurem deste modo a manutenção do bom funcionamento da rede. Estas metodologias basearam-se sobretudo na teoria do controlo usando controladores *Proporcionais* e *Proporcionais-Integradores* para utilização no controlo de admissão de chamadas e assim controlar a carga do sistema.

Após o estudo de várias metodologias, mostrou-se que é possível melhorar o actual controlo implementado no sistema, reduzindo consideravelmente o número de rejeições de chamadas. O melhor desempenho de todas as metodologias estudadas foi obtido com o algoritmo de controlo que utiliza um *Limiar Máximo* de *CPS* e um *Limiar mínimo* Dinâmico calculado com um controlador *PI*.

5.2 Trabalho Futuro

Apenas foi implementado o algoritmo de controlo em *SCF*, no *SCP*. Um próximo passo será implementar este algoritmo no *SDF*. Esta implementação em princípio será mais complicada uma vez que se terá que estudar mais aprofundadamente a relação entre o *SDF* e cada *SCF* e como é que ao certo cada *SCF* influencia o *SDF*. Repare-se que o modelo teórico simulado é apenas um modelo muito simples do modelo real, tendo servido apenas para fazer o estudo e comparação entre diferentes algoritmos de controlo.

Um dos aspectos a ser melhorados neste algoritmo de controlo é a afinação dos k_p e dos k_i . Actualmente esta afinação é feita de forma manual durante testes de carga o que implica sempre algum tempo até encontrar a melhor afinação. Para resolver este problema deverá ser implementado um algoritmo de afinação automático com ajuda dos *logs* do sistema que indiquem valores de rejeição em carga máxima em determinados períodos de tempo.

Outro aspecto que poderá ser melhorado é o balanceamento de carga. Por exemplo, actualmente é definida um limite máximo para cada *SCF* e grupo de *SCF*. Se um dos *SCF*, ou um dos grupos de *SCF* não necessitar de tanta capacidade de carga, esse excedente poderá ser aproveitado por outros grupos e/ou por outros *SCF* aproveitando melhor desta forma os recursos do sistema.

Referências

- [1] LAVRADOR, Alcino - Redes Inteligentes, Portugal Telecom Inovação SA, Aveiro, 1996 (Documento Interno)
- [2] http://en.wikipedia.org/wiki/Intelligent_network (Março 2008)
- [3] A. Folkestad, P. Osland and P. J. Emstad. On Load Control for Service Data Points. 4th International Conference on Intelligence in Networks, November 1996
- [4] A. Menéndez, H. Benitez-Perez and M. A. Palomera-Perez. Node Availability for Distributed Systems considering processor and RAM Utilization. 8th Mexican International Conference on Current Trends in Computer Science table of contents, 2007
- [5] K. D. Kang, J. Oh, Y. Zhou, Backlog Estimation and Management for Real-Time Data Service. 20th Euromicro Conference on Real-Time Systems, July, 2008.
- [6] S. Nadjm-Tehrani, K. Najarian, C. Curescu, T. Lingvall, T. A. Dahlberg, Adaptive Load Control Algorithms for 3rd Generation Mobile Networks, 5th ACM International Workshop on modeling, Analysis and Simulation of Wireless and Mobile Systems.
- [7] MELO, António - Apontamentos de Sistemas de Controlo, DETI, Universidade de Aveiro, 2007
- [8] http://en.wikipedia.org/wiki/PID_controller (Março 2008)
- [9] Redes Inteligentes Avançado, Slides, Portugal Telecom Inovação SA, Aveiro (Documento Interno)
- [10] <http://www.3gpp.org> (Junho 2008)
- [11] <http://www.itu.int> (Junho 2008)

